



Universidad
Carlos III de Madrid

Departamento de Informática

PROYECTO FIN DE CARRERA

Desarrollo de una aplicación para Android destinada al aprendizaje de idiomas

INGENIERÍA TÉCNICA DE INFORMÁTICA DE GESTIÓN

Autor: Alfonso Liñán Carrasco

Tutor: Telmo Zarraonandia Ayo

Leganés, a 21 de Marzo de 2012

Título: Desarrollo de una aplicación para Android destinada al aprendizaje de idiomas
Autor: Alfonso Liñán Carrasco
Director: Telmo Zarraonandia Ayo

EL TRIBUNAL

Presidente: _____

Vocal: _____

Secretario: _____

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día **21** de **Marzo** de **2012** en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE

Agradecimientos

Las siguientes líneas son para agradecer a todas aquellas personas que me han dado su apoyo y que han ejercido una influencia positiva en mí. Gracias a vosotros he conseguido mejorar como persona y poder disfrutar de un futuro mejor.

A mis padres, Pedro y Ana, que siempre me han dado su apoyo incondicional, dándome todo cuanto necesitaba, inculcándome unos valores de esfuerzo y superación que tanto agradezco en estos momentos. Soy lo que soy gracias a vosotros.

A mi tutor Telmo, por darme la oportunidad de realizar este proyecto.

A mis hermanos, Pedro, Javi y Jesús, y mis cuñadas Cristi, Patri y Rebe. Por preocuparse siempre de mí y darme los mejores consejos posibles. Agradecer de forma especial a Javi, por ayudarme siempre que lo he necesitado, porque sin ella, no lo hubiera conseguido, ¡Jugón!

A Alex, por todas aquellas tarde que han marcado este último año, obligándonos a estudiar el uno al otro. Café, Superfighter y biblioteca, ¡así me lo enseñaron a mí! Nunca me habían echado tantas veces de la biblioteca porque tenían que cerrar. Gracias Teo.

A Fer, esa persona que por mucho que le insistas para que se venga a la biblioteca, remolonea y nunca viene. Y eso que estudiamos en la misma universidad. Gracias por los días en la pío.

A Borja y Saba, por preguntarme siempre que tal iba, por buscar trabajos que nos harán millonarios y por el piso a lo “friends” que nos vamos a montar.

A Yoli y Germán, por los desayunos mañaneros inolvidables, por los bocadillos de oreja y todos los cafés para no dormirme. Parte indispensable y necesaria en mi día a día universitario.

En definitiva, a todas aquellas personas que han influido en mí durante mi vida universitaria, compañeros, amigos y profesores.

Resumen

Este proyecto muestra el desarrollo de una aplicación para teléfonos inteligentes con sistema operativo Android, destinada al aprendizaje de idiomas.

El objetivo de MobileTestAndroidPlayer es dar al usuario un software capaz de mostrar los materiales educativos creados por los educadores de lenguas extranjeras para fomentar el aprendizaje de idiomas.

El documento ofrece una visión del contexto general en el que se desenvuelve la aplicación, llamado m-learning y los teléfonos inteligentes. Además, se incluirá en el documento el desarrollo de la aplicación para la plataforma que abarca las fases de análisis, diseño e implementación dentro del ciclo de vida del software.

Palabras clave: Android, Smartphone, m-learning, aplicación, test, análisis, diseño, requisitos, casos de uso, diagramas, implementación.

Abstract

This project describes the development of an application for Smartphone with Android OS, intended for language learning.

The goal of MobileTestAndroidPlayer is to give the user a software able to display the educational materials created by educators from foreign language to encourage language learning.

The document provides an overview of the general context in which the application is developed, called m-learning and smartphones. Furthermore, the development of an application will be included within the document that will include the analysis, design and implementation phases of software lifecycle.

Keywords: Android, Smartphone, m-learning, application, test, analysis, design, requirement, use case, diagrams, implementation.

Índice general

INTRODUCCIÓN Y OBJETIVOS.....	14
1.1 INTRODUCCIÓN.....	15
1.2 OBJETIVOS.....	17
1.3 FASES DEL DESARROLLO	17
1.4 MEDIOS EMPLEADOS	18
1.5 ESTRUCTURA DE LA MEMORIA.....	19
ESTADO DEL ARTE	21
2.1 DISPOSITIVOS MÓVILES.....	22
2.1.1 Introducción	22
2.1.2 Smartphone.....	23
2.2 SISTEMAS OPERATIVOS	24
2.2.1 Android.....	25
2.2.2 IOS.....	25
2.2.3 Symbian OS	26
2.2.4 Blackberry OS	27
2.2.5 Windows Phone.....	27
2.3 JAVA.....	28
2.4 XML	29
2.5 JUSTIFICACIÓN	30
LA PLATAFORMA ANDROID.....	35
3.1 CARACTERÍSTICAS.....	36
3.2 ARQUITECTURA.....	36
3.3 LA MÁQUINA VIRTUAL DALVIK	37
3.4 INSTALACIÓN DEL ENTORNO Y KIT DE DESARROLLO.	38
3.5 INTERFAZ DE USUARIO	38
3.6 ESTRUCTURA DE UN PROYECTO ANDROID EN ECLIPSE	39
3.7 PRIMER ARRANQUE DEL SDK DE ANDROID	40
3.8 PRIMERA APLICACIÓN EN ANDROID: “HOLA MUNDO”	41
ANÁLISIS	45

4.1 INTRODUCCIÓN A MOBILE TEST ANDROID PLAYER	46
4.2 ARQUITECTURA GENERAL DEL SISTEMA	47
4.3 CASOS DE USO	48
4.4 REQUISITOS DEL SISTEMA	63
4.4.1. <i>Requisitos Funcionales</i>	66
4.4.2. <i>Requisitos No Funcionales</i>	75
DISEÑO	81
5.1 DIAGRAMA DETALLADO	82
5.2 DIAGRAMAS DE SECUENCIA	88
5.3 PROTOTIPOS DE LA INTERFAZ DE USUARIO	96
5.3.1 <i>Pantalla Menú Principal</i>	96
5.3.2 <i>Pantalla Descarga Material Educativo</i>	97
5.3.3 <i>Pantalla Listar Material Educativo</i>	98
5.3.4 <i>Pantallas de Ejercicios</i>	99
5.3.5 <i>Pantalla de Soluciones y Puntuación</i>	100
IMPLEMENTACIÓN	101
6.1 DESCARGA MATERIAL EDUCATIVO	102
6.2 LECTURA DE LOS ARCHIVOS XML	104
6.3 LISTADOS DE CONTENIDOS	107
6.4 FUNCIONALIDADES EN LOS EJERCICIOS	108
6.5 PRUEBAS	110
6.5.1 <i>Descarga Material Educativo</i>	110
6.5.2 <i>Lectura de los Archivos XML</i>	111
6.5.3 <i>Listados de Contenidos</i>	111
6.5.4 <i>Funcionalidades de los Ejercicios</i>	112
MANUAL DE USUARIO Y DE INSTALACIÓN	115
7.1 MANUAL DE USUARIO	116
7.2 MANUAL DE INSTALACIÓN	129
CONCLUSIONES Y TRABAJOS FUTUROS	132
8.1 CONCLUSIONES	133
8.2 TRABAJOS FUTUROS	134
GESTIÓN DEL PROYECTO	135
9.1 PLANIFICACIÓN	136
9.2 PRESUPUESTO	140
MOBILE TEST CREATOR	142
REFERENCIAS	145

Índice de figuras

FIGURA 1. CUOTA MUNDIAL DE MERCADO DE SISTEMAS OPERATIVOS MÓVILES [12]	24
FIGURA 2. CUOTA DE SISTEMAS OPERATIVOS PARA DISPOSITIVOS MÓVILES EN ESPAÑA [12]	30
FIGURA 3. CUOTA DE SISTEMAS OPERATIVOS PARA DISPOSITIVOS MÓVILES EN EL MUNDO [12]	31
FIGURA 4. NÚMERO TOTAL DE APLICACIONES EN LOS MARKETS [24]	32
FIGURA 5. PRECIO MEDIO DE LAS APPS [25]	33
FIGURA 6. ARQUITECTURA INTERNA DE ANDROID [27]	37
FIGURA 7. JERARQUÍA QUE DEFINE UNA INTERFAZ DE USUARIO [28]	38
FIGURA 8. ESTRUCTURA DE UN PROYECTO ANDROID	39
FIGURA 9. ESTRUCTURA CARPETA RESC DE UN PROYECTO ANDROID	40
FIGURA 10. BARRA SUPERIOR DE ICONOS DE ECLIPSE	44
FIGURA 11. VENTANA DEL EMULADOR SDK DE ANDROID [30]	44
FIGURA 12. DIAGRAMA GENERAL DEL SISTEMA	47
FIGURA 13. DIAGRAMA DE CASOS DE USO	50
FIGURA 13. DIAGRAMA DETALLADO DE MOBILETESTANDROIDPLAYER	82
FIGURA 14. DIAGRAMA DE COMPONENTES DE MOBILETESTANDROIDPLAYER	83
FIGURA 15. DIRECTORIO ALMACENAMIENTO MATERIAL EDUCATIVO	87
FIGURA 16. DIAGRAMA DE SECUENCIA VER INSTRUCCIONES	88
FIGURA 17. DIAGRAMA DE SECUENCIA IMPORTAR NUEVO MATERIAL	89
FIGURA 18. DIAGRAMA DE SECUENCIA SELECCIONAR MATERIAL	89
FIGURA 19. DIAGRAMA DE SECUENCIA NAVEGAR POR EL TEST	90
FIGURA 20. DIAGRAMA DE SECUENCIA VER ANOTACIONES	90
FIGURA 21. DIAGRAMA DE SECUENCIA CONSULTAR PUNTUACIÓN	91
FIGURA 22. DIAGRAMA DE SECUENCIA CONSULTAR SOLUCIONES	91
FIGURA 23. DIAGRAMA DE SECUENCIA CORREGIR TEST AUDIO CHOICE	92
FIGURA 24. DIAGRAMA DE SECUENCIA CORREGIR TEST DRAG AND DROP	92
FIGURA 25. DIAGRAMA DE SECUENCIA CORREGIR TEST MULTIPLE GAP	92
FIGURA 26. DIAGRAMA DE SECUENCIA CORREGIR TEST WORD ORDER	93
FIGURA 27. DIAGRAMA DE SECUENCIA ESCUCHAR AUDIO	93
FIGURA 28. DIAGRAMA DE SECUENCIA ESCUCHAR AUDIO	94
FIGURA 29. DIAGRAMA DE SECUENCIA REINICIAR TEST	94
FIGURA 30. DIAGRAMA DE SECUENCIA SALIR MENÚ PRINCIPAL	95
FIGURA 31. DIAGRAMA DE SECUENCIA SALIR	95
FIGURA 32. PROTOTIPO PANTALLA MENÚ PRINCIPAL	96

FIGURA 33. PROTOTIPO PANTALLA DESCARGA MATERIAL EDUCATIVO.....	97
FIGURA 34. PROTOTIPO PANTALLA LISTAR MATERIAL EDUCATIVO	98
FIGURA 35. PROTOTIPO PANTALLAS DE EJERCICIOS.....	99
FIGURA 36. PROTOTIPO PANTALLA DE SOLUCIONES Y PUNTUACIÓN	100
FIGURA 37. PANTALLA PRINCIPAL MOBILETESTANDROIDPLAYER.....	116
FIGURA 38. PANTALLA LISTADO DE MATERIALES EDUCATIVOS.....	116
FIGURA 39. PANTALLAS DESCARGA MATERIAL EDUCATIVO.....	117
FIGURA 40. PANTALLA DE ERROR	118
FIGURA 41. PANTALLA INSTRUCCIONES.....	118
FIGURA 42. PANTALLA SELECCIONAR MATERIAL Y EMPEZAR.....	119
FIGURA 43. PANTALLA BORRAR MATERIAL.....	119
FIGURA 44. MENÚ INFERIOR DE NAVEGACIÓN	120
FIGURA 45. MENSAJE DE ERROR SIN SELECCIÓN.....	120
FIGURA 46. PANTALLA INTRODUCTION	121
FIGURA 47. PANTALLAS THEORY	122
FIGURA 48. PANTALLA THEORY EXPANDIR CONTENIDO.....	122
FIGURA 49. PANTALLA THEORY ANOTACIONES.....	123
FIGURA 50. PANTALLA THEORY ANOTACIONES + EXPANDIR.....	123
FIGURA 51. PANTALLA FILL THE GAP.....	124
FIGURA 52. PANTALLA WORD ORDER	124
FIGURA 53. PANTALLA AUDIO.....	125
FIGURA 54. PANTALLA AUDIO CHOICE.....	125
FIGURA 55. PANTALLA DRAG AND DROP	126
FIGURA 56. PANTALLA MULTIPLE CHOICE	126
FIGURA 57. PANTALLA MULTIPLE FILL THE GAP	127
FIGURA 58. PANTALLA MENÚ SCORE, SOLUTION AND RESET.....	127
FIGURA 59. PANTALLA SCORE.....	128
FIGURA 60. PANTALLA SOLUTION.....	128
FIGURA 61. PANTALLA PRINCIPAL.....	129
FIGURA 62. PANTALLA ADMINISTRADOR DE ARCHIVOS.....	130
FIGURA 63. PANTALLA ARCHIVO MOBILETESTANDROIDPLAYER.APK	130
FIGURA 64. PANTALLA MENÚ TELÉFONOS ANDROID	131
FIGURA 65. ESTIMACIÓN INICIAL: FECHAS Y DURACIÓN.....	136
FIGURA 66. GRÁFICA GANTT ESTIMACIÓN INICIAL.....	137
FIGURA 67. ESTIMACIÓN REAL: FECHAS Y DURACIÓN	138
FIGURA 68. GRÁFICA GANTT ESTIMACIÓN REAL	139
FIGURA 69. PANTALLA CREACIÓN DE PÁGINAS DE MOBILETESTCREATOR	142
FIGURA 70. PANTALLA CREACIÓN DE TEST DE MOBILETESTCREATOR	143

Índice de tablas

TABLA 01. TABLA CASOS DE USO	49
TABLA 02. TABLA CASO DE USO VER INSTRUCCIONES.....	51
TABLA 03. TABLA CASO DE USO IMPORTAR NUEVO MATERIAL	52
TABLA 04. TABLA CASO DE USO SELECCIONAR MATERIAL EDUCATIVO	53
TABLA 05. TABLA CASO DE USO SALIR.....	54
TABLA 06. TABLA CASO DE USO NAVEGAR POR EL TEST	55
TABLA 07. TABLA CASO DE USO VER ANOTACIONES	56
TABLA 08. TABLA CASO DE USO CONSULTAR PUNTUACIÓN	57
TABLA 09. TABLA CASO DE USO CONSULTAR SOLUCIONES	58
TABLA 10. TABLA CASO DE USO CORREGIR TEST	59
TABLA 11. TABLA CASO DE USO ESCUCHAR AUDIO	60
TABLA 12. TABLA CASO DE USO REINICIAR TEST	61
TABLA 13. TABLA CASO DE USO SALIR AL MENÚ PRINCIPAL.....	62
TABLA 14. TABLA SIGLAS REQUISITOS	64
TABLA 15. PLANTILLA DE REQUISITOS	65
TABLA 16. REQUISITO FUNCIONAL 01	66
TABLA 17. REQUISITO FUNCIONAL 02	66
TABLA 18. REQUISITO FUNCIONAL 03	67
TABLA 19. REQUISITO FUNCIONAL 04	67
TABLA 20. REQUISITO FUNCIONAL 05	67
TABLA 21. REQUISITO FUNCIONAL 06	68
TABLA 22. REQUISITO FUNCIONAL 07	68
TABLA 23. REQUISITO FUNCIONAL 08	68
TABLA 24. REQUISITO FUNCIONAL 09	69
TABLA 25. REQUISITO FUNCIONAL 10	69
TABLA 26. REQUISITO FUNCIONAL 11	69
TABLA 27. REQUISITO FUNCIONAL 12	70
TABLA 28. REQUISITO FUNCIONAL 13	70
TABLA 29. REQUISITO FUNCIONAL 14	70
TABLA 30. REQUISITO FUNCIONAL 15	71
TABLA 31. REQUISITO FUNCIONAL 16	71
TABLA 32. REQUISITO FUNCIONAL 17	71

TABLA 33. REQUISITO FUNCIONAL 18	72
TABLA 34. REQUISITO FUNCIONAL 19	72
TABLA 35. REQUISITO FUNCIONAL 20	72
TABLA 36. REQUISITO FUNCIONAL 21	73
TABLA 37. REQUISITO FUNCIONAL 22	73
TABLA 38. REQUISITO FUNCIONAL 23	73
TABLA 39. REQUISITO FUNCIONAL 24	74
TABLA 40. REQUISITO FUNCIONAL 25	74
TABLA 41. REQUISITO DE INTERFAZ 01	75
TABLA 42. REQUISITO DE INTERFAZ 02	75
TABLA 43. REQUISITO DE INTERFAZ 03	75
TABLA 44. REQUISITO DE INTERFAZ 04	76
TABLA 45. REQUISITO DE INTERFAZ 05	76
TABLA 46. REQUISITO DE INTERFAZ 06	76
TABLA 47. REQUISITO DE INTERFAZ 07	77
TABLA 48. REQUISITO DE USABILIDAD 01	77
TABLA 49. REQUISITO DE USABILIDAD 02	77
TABLA 50. REQUISITO DE USABILIDAD 03	78
TABLA 51. REQUISITO DE RENDIMIENTO 01	78
TABLA 52. REQUISITO DE RENDIMIENTO 02	78
TABLA 53. REQUISITO DE RENDIMIENTO 03	79
TABLA 54. REQUISITO OPERACIONAL Y AMBIENTAL 01	79
TABLA 55. REQUISITO DE SOPORTE 01	79
TABLA 56. REQUISITO DE SOPORTE 02	80
TABLA 57. REQUISITO DE SEGURIDAD 01	80
TABLA 58. REQUISITO DE SEGURIDAD 02	80
TABLA 59. COMPONENTE INTERFAZ DE USUARIO	83
TABLA 60. COMPONENTE GESTIÓN DE PAQUETES	84
TABLA 61. COMPONENTE INTRODUCTION	84
TABLA 62. COMPONENTE THEORY	84
TABLA 63. COMPONENTE AUDIO	84
TABLA 64. COMPONENTE AUDIO CHOICE	85
TABLA 65. COMPONENTE WORD ORDER	85
TABLA 66. COMPONENTE MULTIPLE GAP	85
TABLA 67. COMPONENTE DRAG AND DROP	86
TABLA 68. COMPONENTE DOWNLOAD	86
TABLA 69. COMPONENTE UNZIP	86
TABLA 70. COMPONENTE SD CARD	87
TABLA 71. COMPONENTE LECTURA DE PAQUETES	88
TABLA 72. TABLA DE PRUEBAS	110
TABLA 73. PRUEBA DESCARGAR MATERIAL EDUCATIVO	110
TABLA 74. PRUEBA LECTURA ARCHIVOS XML	111
TABLA 75. PRUEBA LISTADO DE CONTENIDOS	111
TABLA 76. PRUEBA CORREGIR EJERCICIO	112
TABLA 77. PRUEBA TEXTO CON ESTILOS HTML	112
TABLA 78. PRUEBA SIGUIENTE/ANTERIOR EJERCICIO O GAP	113
TABLA 79. PRUEBA REPRODUCIR AUDIO	113
TABLA 80. PRUEBA EXPANDIR TEORÍA	113
TABLA 81. PRUEBA INTRODUCIR TEXTO	114
TABLA 82. PRUEBA MOSTRAR SOLUCIONES/PUNTUACIONES	114
TABLA 83. AMORTIZACIÓN DE EQUIPOS	140
TABLA 84. RESUMEN COSTES	141
TABLA 85. GLOSARIO DE TÉRMINOS	144

Índice de código

CÓDIGO 01. EJEMPLO SIMPLE CON TEXTO Y UN BOTÓN.	39
CÓDIGO 02. CÓDIGO JAVA DEL ARCHIVO HOLAMUNDO.JAVA DE ECLIPSE	42
CÓDIGO 03. CÓDIGO JAVA DEL ARCHIVO HOLAMUNDO.JAVA DE ECLIPSE	43
CÓDIGO 04. CÓDIGO XML DEL ARCHIVO MAIN.XML DE ECLIPSE [30]	43
CÓDIGO 05. CÓDIGO FUNCIÓN DOWNLOADFROMURL.....	102
CÓDIGO 06. CÓDIGO FUNCIÓN UNZIP	103
CÓDIGO 07. CÓDIGO DEL THREAD PARA DESCARGARMATERIAL EDUCATIVO	103
CÓDIGO 08. CÓDIGO DEL THREAD PARA DESCOMPRIMIR MATERIAL EDUCATIVO	104
CÓDIGO 09. CÓDIGO XML DEL ARCHIVO TESTCOLLECTION	104
CÓDIGO 10. CÓDIGO DEL PARSER DEL ARCHIVO TESTCOLLECTION.....	105
CÓDIGO 11. CÓDIGO DEL PARSER DEL ARCHIVO TESTCOLLECTION.....	106
CÓDIGO 12. CÓDIGO DEL PARSER DEL ARCHIVO PAGECOLLECTION	106
CÓDIGO 13. CÓDIGO ARRAYADAPTER	107
CÓDIGO 14. CÓDIGO BORRAR MATERIAL EDUCATIVO	108
CÓDIGO 15. CÓDIGO BOTÓN REPRODUCIR AUDIO.....	108
CÓDIGO 16. CÓDIGO CONTROL DE VOLUMEN.....	109
CÓDIGO 17. CÓDIGO DEL ARCHIVO GLOBALES	109

Capítulo 1

INTRODUCCIÓN Y OBJETIVOS

1.1 Introducción

Por definición, el **e-Learning** es “el suministro de programas educativos y sistemas de aprendizaje a través de medios electrónicos” (Mendoza, 2003, párr. 2). El **e-Learning** está basado en el uso de un dispositivo móvil como por ejemplo un ordenador o un teléfono móvil, para proveer a las personas de material educativo. La base del desarrollo del **e-learning** es la educación a distancia, que también resuelve las dificultades del alumno en cuanto a desplazamientos, solapamientos de horarios y los problemas habituales de una enseñanza presencial. [1]

Dentro de las múltiples ventajas que ofrece el e-Learning, probablemente las dos más importantes pueden ser: [2]

- Ahorra costos y desplazamiento.
- Favorece una mejor formación del estudiante debido al amplio volumen de información, su nivel de actualización, la autonomía del estudiante y su flexibilidad de horarios. En resumen, una formación “*Just in time, just for me*”: lo que el alumno quiere, cuando el alumno lo quiere.

En el caso de los inconvenientes, a continuación se presentan las dos más importantes:

- La voluntad del alumno es una condición *sine qua non* ya que sin ella el entrenamiento es imposible. Si el alumno no tiene voluntad esta ventaja de flexibilidad en el uso tanto del tiempo como del lugar y el horario se tornarán en desventajas.
- Depende de la conexión a Internet y de la existencia de un ordenador, lo que conlleva que, si es un equipo personal, debe hacer frente a los costes de acceso a Internet, y si no es personal y se acude a un centro de ordenadores, debe desplazarse y respetar los horarios establecidos.

El uso de las tecnologías móviles se viene incorporando desde hace tiempo a nuestra vida cotidiana. Dicho uso está derivando el **e-Learning** a un nuevo estilo de educación a distancia llamado **mobile learning** o **m-learning**.

“El **m-learning**, es una evolución del **e-learning**. Significa **aprendizaje electrónico móvil** o educación móvil y es, como su propio nombre indica, una metodología o difusión de contenidos de enseñanza y aprendizaje a través del uso de pequeños y maniobrables dispositivos móviles, tales como Smartphones, tablets, agendas electrónicas, i-pods y todo dispositivo de mano que tenga alguna forma de conectividad inalámbrica, es decir, que no tenga cables” (Rubio Gómez, Celia. González moreno, Vanessa. Selas Felguera, M^a José. s.f. párr. 2). [3]

Las Principales ventajas del m-learning son: [4]

- Los dispositivos móviles pueden ser usados en cualquier parte y en cualquier momento, por lo que el proceso de aprendizaje se personaliza y adapta a los requerimientos y disponibilidades individuales del estudiante.
- Cada día que hay más lugares desde los que es posible acceder a Internet mediante una conexión WIFI o Wireless Fidelity, por tanto el usuario no tiene la necesidad de contratar una tarifa de datos.

En la otra cara de la moneda encontramos que las limitaciones del **m-learning** son, sobre todo, las mismas que las inherentes a los propios dispositivos móviles:

- Debido a su menor tamaño los dispositivos móviles suelen ser por lo general menos cómodos para trabajar que un ordenador personal. Por tanto, hay que tener en cuenta la dificultad de adaptar los contenidos educativos a este tipo de aparatos.
- La principal barrera del **m-Learning** es la conexión a Internet. Los Smartphone poseen acceso a redes wifi, permitiendo un acceso a contenidos on-line. Pero no siempre se puede disfrutar de conexión inalámbrica, siendo algo que se antoja indispensable la contratación de una tarifa de datos por parte del usuario. Lo que supone un coste para el usuario, aunque se está abaratando notablemente, mantener una conexión desde este tipo de dispositivos no es barato. Dicho coste adicional reduce el número de usuarios potenciales para acceder al **m-learning**. Así pues, hacen falta aplicaciones que se puedan utilizar sin necesidad de conexión a Internet. [5]
- La última limitación inherente a los dispositivos móviles es la fragmentación del mercado en cuanto al sistema operativo residente en el Smartphone. Actualmente, son cinco los sistemas operativos que se reparten el grueso del mercado, siendo estos, Android, IOS, Symbian OS, BlackBerry OS y Windows Phone. Esto supone que para que una aplicación soporte los cinco sistemas, hemos tenido que diseñar cinco veces la misma aplicación y trabajar en cinco plataformas diferentes y con cinco lenguajes de programación distintos.

A estos inconvenientes hay que sumarle otra desventaja que se puede considerar propia del **e-Learning** en general y es que, en la mayoría de los casos, el educador que desee crear material educativo en formato digital debe o bien contar con formación específica en este campo o bien encargar a expertos en el área la producción de dicho material. Es necesario, por tanto, proporcionar a los educadores herramientas sencillas de usar que oculten la complejidad de la programación y permitan de forma sencilla y rápida crear material educativo digital de calidad y optimizado para su uso en dispositivos móviles.

1.2 Objetivos

El presente trabajo se engloba dentro de un proyecto que tiene por finalidad aliviar los problemas previamente mencionados proporcionando por un lado, a los profesores de lengua extranjera una herramienta que permita la fácil generación de material educativo, y por otro lado a sus alumnos un software capaz de desplegar dicho material en sus móviles sin necesidad de estar conectados a la red. Más específicamente este proyecto se centrará en esta segunda parte.

El objetivo principal es por tanto crear una aplicación o *player* para Smartphones con sistema operativo Android que lea las descripciones del material educativo generados por el profesor y las muestre al alumno con un formato u otro, dependiendo del tipo de material que sea. Estas descripciones serán proporcionadas por medio de ficheros XML que el programa deberá interpretar.

La funcionalidad básica de la aplicación deberá posibilitar:

- Importar colecciones de material desde una determinada dirección Web.
- Guardar y leer las colecciones de ejercicios desde la tarjeta sd del Smartphone.
- Dar soporte a la ejecución de ejercicios y test, permitiendo al alumno conocer el número de respuestas acertadas, la solución correcta y repetir ejercicio.

La ejecución de todos los tipos de material que la aplicación del profesor permite crear, entre los que se encuentran introducciones, material teórico, ejercicios de rellenar hueco, de elección múltiple, ejercicios de listening y de arrastrar y soltar.

1.3 Fases del desarrollo

Consideraremos las siguientes fases del desarrollo para nuestro proyecto, basadas en el modelo en cascada, es decir, de forma secuencial: [6]

- Análisis de requisitos

Extraer los requisitos debe ser el primer paso. El desarrollador debe comprender la naturaleza del problema, las necesidades del cliente, la función requerida así como el comportamiento de la aplicación a realizar. Para ello se debe desarrollar un catálogo de requisitos que recoja dichos aspectos.

- Diseño

Determinar el comportamiento esperado de la aplicación, es decir, el funcionamiento básico sin entrar en detalles.

- Implementación

Reducir el diseño a código. La complejidad y la duración la determinará en mayor o menos medida el lenguaje de programación utilizado.

- Pruebas

Comprobar que el software realice correctamente las tareas indicadas en el diseño y del catálogo de requisitos.

- Documentación

Modelaciones (UML), diagramas, implementación, pruebas, manuales de usuario, manuales técnicos...

1.4 Medios empleados

- Entorno de desarrollo integrado de código abierto multiplataforma Eclipse.
- Kit de desarrollo de software o SDK de Android.
- Java development kit.
- Netbook Hp Mini 110-3120ss.
- Mobile Test Creator. (referencias a esta herramienta en el ANEXO A)
- Servicio de alojamiento de archivos multiplataforma Dropbox.
- Smartphone Orange Boston.
- Pencil para la creación de prototipos.
- Microsoft Visio 2003 para la creación de diagramas de secuencia.
- GanttProject para la creación del diagrama de Gantt.

1.5 Estructura de la memoria

En el presente apartado se explica de forma breve y concisa el contenido de cada capítulo y anexo del documento:

- El **capítulo I** nos adentra en el contexto general en el que se envuelve la aplicación, explicando el problema existente y la solución propuesta para dicho problema. Termina mostrando la estructura del presente documento.
- El **capítulo II** nos acerca la actualidad de la telefonía móvil y de los sistemas operativos móviles más usados, para mostrar una visión general y posteriormente justificar la elección de la plataforma deseada.
- En el **capítulo III** se muestra un resumen de la plataforma elegida, Android, material para introducirnos en el mundo en el que se desarrolla la aplicación.
- En el **capítulo IV** se encuentra la fase de análisis donde podrá encontrar aspectos clave para el buen desarrollo del producto como una introducción a la aplicación y los estudios sobre la arquitectura utilizada, casos de uso y requisitos.
- El **capítulo V**, diseño, determina el comportamiento esperado de la aplicación mediante una serie de diagramas de secuencia y un prototipo de bajo nivel.
- El **capítulo VI** ahonda en el desarrollo del sistema, es decir, como se ha conseguido llevar a cabo la implementación de las funcionalidades de la aplicación. Además incluye las pruebas realizadas para determinar el buen funcionamiento del mismo.
- El **capítulo VII** contiene el manual de usuario para el buen manejo de la aplicación, así como el manual de instalación para aquellos usuarios que no estén tan familiarizados con la plataforma Android.
- El **capítulo VIII** presenta las conclusiones obtenidas una vez finalizado el desarrollo del proyecto, y una serie de trabajos futuros que podrían aplicarse como mejoras de la aplicación.
- En el **capítulo IX** se encuentra la planificación del proyecto y un presupuesto del coste real que hubiera supuesto.
- Por último, se incluyen tres anexos en el presente documento:
 - **Anexo A: Mobile Test Creator**, donde podemos encontrar una pequeña explicación sobre la utilidad de la herramienta en el proyecto.

- **Glosario**, que incluye los términos utilizados en el desarrollo de la documentación para una mejor comprensión.
- **Referencias**, que incluye toda documentación consultada para la elaboración del proyecto.

Capítulo 2

ESTADO DEL ARTE

2.1 Dispositivos Móviles

2.1.1 Introducción

Los **dispositivos móviles** son “aparatos de pequeño tamaño, con algunas capacidades de procesamiento, con conexión permanente o intermitente a una red, con memoria limitada, diseñados específicamente para una función, pero que pueden llevar a cabo otras funciones más generales.” (Baz Alonso, Ferreira Artime, Álvarez Rodríguez, García Baniello, s.f., párr. 2).

Cada dispositivo móvil tiene un número de funcionalidades distinto al de sus compañeros, por esta razón, **T38** y **DuPont Global Mobility Innovation Team** en el año 2005 propusieron la siguiente clasificación de dispositivos móviles: [7]

- **Dispositivo Móvil de Datos Limitados** (*Limited Data Mobile Device*): Referidos a dispositivos móviles de pantalla pequeña de tipo texto, donde los servicios de datos suelen estar limitados al envío y recepción de mensajes de texto y conexiones de tipo WAP.
- **Dispositivo Móvil de Datos Básicos** (*Basic Data Mobile Device*): Referidos a dispositivos móviles con una pantalla de tamaño medio y una navegación basada en iconos. Además ofrecen envío y recepción de e-mails, mensajes de texto y navegador web entre otras funcionalidades. Este grupo lo forman los terminales llamados **Smartphone**.
- **Dispositivo Móvil de Datos Mejorados** (*Enhanced Data Mobile Device*): Referidos a dispositivos móviles con una pantalla de mediana a grande que ofrecen la misma funcionalidad que los **Dispositivos Móviles de Datos Básicos** además de aplicaciones de Microsoft como pueden ser el Word y Excel entre otras.

De entre los elementos que pueden llegar a ser un dispositivo móvil (una PDA, un teléfono móvil, un lector de libros electrónico o un ordenador portátil) destaca, sin lugar a dudas, el teléfono móvil como el dispositivo más utilizado de entre todos. No obstante, según los datos de la Comisión del Mercado de Telecomunicaciones a fecha de Mayo del 2011, en España la media de teléfonos móviles por persona es superior a un teléfono por habitante (117,2 líneas por cada 100 habitantes) [8]. Si vamos más allá de nuestras fronteras, y nos fijamos en los datos a nivel mundial del número de líneas móviles, comprobamos que hay estimado un total de 6.000 millones de líneas [9].

2.1.2 Smartphone

Un Smartphone (cuya traducción sería “teléfono inteligente”) “es una evolución del teléfono móvil tradicional que cuenta con ciertas características y prestaciones que lo acercan más a un ordenador personal que a un teléfono tradicional.” (Aranaz Tudela. 2009) [10]

Entre dichas prestaciones y características, se encuentran una amplia mejora del almacenamiento de datos, conexión a Internet mediante una tarifa contratada o haciendo uso de redes WIFI, acelerómetro, pantalla táctil, teclado QWERTY...y un sinfín de aplicaciones de usuario, además de la posibilidad de descarga de nuevas aplicaciones.

Todas estas prestaciones y características de los Smartphone estarían desaprovechadas sin software que las saque partido. Por ello los Smartphone llevan un SO que les permite realizar todas estas tareas de una forma rápida y sencilla.

En el siguiente apartado, realizaremos una breve descripción de los SO para dispositivos móviles y Smartphone más importantes que se encuentran en el mercado a día de hoy.

2.2 Sistemas Operativos

Partiendo de la definición de **sistema operativo**: “Capa compleja entre el hardware y el usuario, concebible también como una máquina virtual, que facilita al usuario o al programador las herramientas e interfaces adecuadas para realizar sus tareas informáticas, abstrayéndole de los complicados procesos necesarios para llevarlas a cabo.” (Baz Alonso, Ferreira Artime, Álvarez Rodríguez, García Baniello, s.f.). [11]

Por ende, un **sistema operativo móvil** es un **sistema operativo** que controla un dispositivo móvil. Sin embargo, estos son mucho más simples, hay más variedad con respecto a los **sistemas operativos** de un ordenador, y el uso de un **sistema operativo** u otro determinará la funcionalidad y la forma de interactuar con el usuario.

La popularidad de los Smartphone aumenta día tras día, y con ella, la importancia de los **sistemas operativos** con los que funcionan. Actualmente, según estudios realizados por **Statcounter**, la cuota de mercado de sistemas operativos móviles en el año 2011 se divide de la siguiente forma:

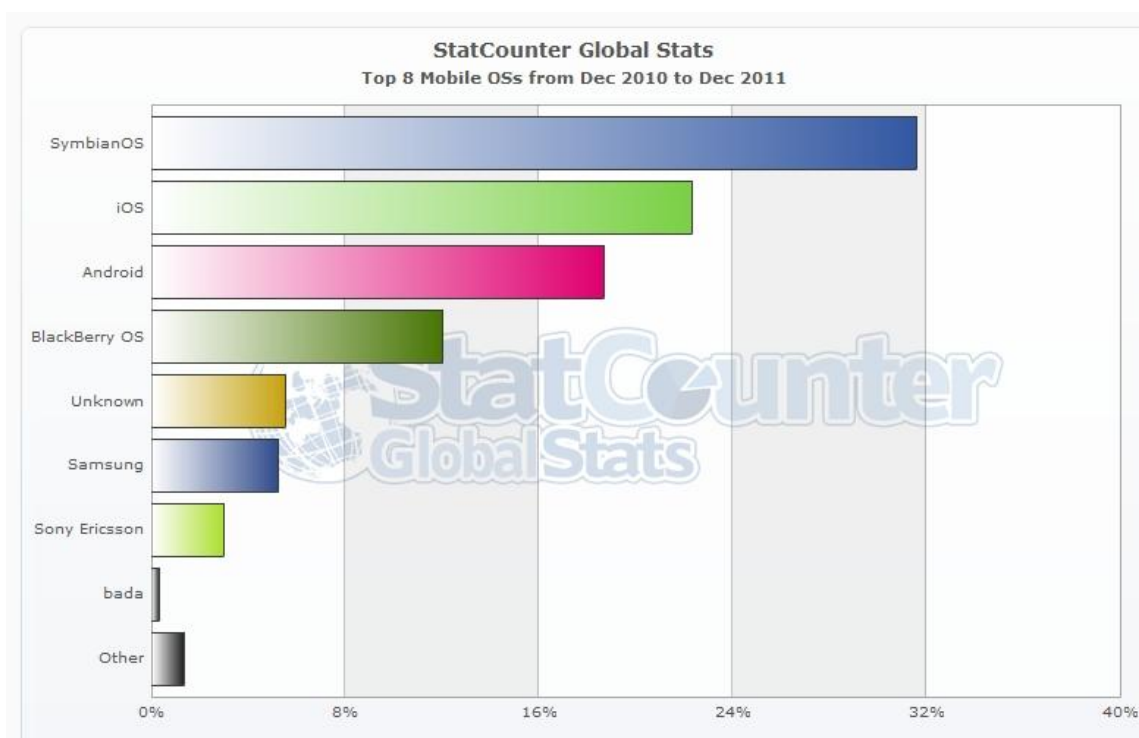


Figura 1. Cuota mundial de mercado de sistemas operativos móviles [12]

En los siguientes apartados ahondaremos en los sistemas operativos más usados.

2.2.1 Android

Android es un sistema operativo basado en la plataforma software de Linux y como éste, libre, gratuito y multiplataforma. Está diseñado para su uso en Smartphone aunque actualmente también está disponible para Tablets. Mediante el uso de bibliotecas desarrolladas con el lenguaje de programación Java, obtenemos el control del dispositivo móvil.

Android es una plataforma de **código abierto**, esto quiere decir que cualquier desarrollador puede crear aplicaciones escritas en un lenguaje de programación y compilarlas después a código nativo de ARM (API de Android).

Inicialmente, **Android** fue desarrollada por **Google Inc.** aunque poco después se unió **Open Handset Alliance**. **Google** sin embargo, ha sido quien ha publicado la mayoría del código fuente de Android bajo la licencia de Software Apache, una licencia de software libre y de código abierto a cualquier desarrollador. [13]

Como principales características y especificaciones podemos destacar:

- **Diseño de dispositivo:** La plataforma es adaptable a cualquier tipo de pantalla.
- **Máquina virtual Dalvik:** Base de llamadas de instancias, muy parecida a Java y diseñada específicamente para Android.
- **Entorno de desarrollo:** Gratuito.
- **Market:** Catálogo de aplicaciones para descargas en dispositivos Android sin necesidad de un PC.
- **Multitarea:** Las aplicaciones en segundo plano reciben ciclos de reloj, a diferencia de otros sistemas.
- **Tethering:** Uso del terminal como punto de acceso inalámbrico.

2.2.2 IOS

IOS es el sistema operativo móvil de Apple. Es una versión reducida de Mac OS X. El **IOS** consta de 4 capas de abstracción:

- Capa del núcleo del sistema operativo,
- Capa de Servicios Principales,
- Capa de Medios de comunicación
- Capa de Cocoa Touch.

Actualmente se encuentran en su quinta versión, que tiene como principales características y especificaciones: [14]

- **Interfaz de usuario:** interfaz multi-Touch. La interacción con el sistema operativo se realiza mediante gestos como deslizar, tocar y pellizcar.
- **Multitarea:** A partir de iOS 4. Sin embargo, no consiste en una verdadera multitarea, pues las aplicaciones ajenas al SO, quedan congeladas en segundo plano no recibiendo un solo ciclo de reloj del procesador.
- **Diseño sobre seguro:** Todas las apps se ejecutan en un entorno seguro, así que ningún sitio web ni ninguna aplicación pueden acceder a los datos de otras apps. Como iOS es compatible con la comunicación en red cifrada, tu información confidencial siempre está protegida. Para salvaguardar tu privacidad, las apps que soliciten información sobre tu ubicación deberán contar con tu consentimiento previo.
- **Accesibilidad integrada:** iOS incorpora numerosas prestaciones de accesibilidad para que las personas con discapacidad disfruten de todo lo que ofrece el iPhone. Por ejemplo, la tecnología integrada de lectura de pantalla VoiceOver permite a los invidentes y a las personas con visión reducida oír una descripción del elemento que están tocando en la pantalla. Además iOS también es compatible de serie con más de 30 teclados braille inalámbricos y ofrece otras muchas prestaciones de accesibilidad, como el aumento dinámico de la pantalla, subtítulos ocultos, sonido mono, texto blanco sobre negro, etc.

2.2.3 Symbian OS

“**Symbian OS** es un sistema operativo que fue producto de la alianza de varias empresas de telefonía móvil, entre las que se encuentran Nokia, Sony Ericsson, PSION, Samsung, Siemens, Arima, Benq, Fujitsu, Lenovo, LG, Motorola, Mitsubishi Electric, Panasonic, Sharp, etc. Sus orígenes provienen de su antepasado EPOC32, utilizado en PDA's y Handhelds de PSION.” (W-Bot, 2005, párr. 1) [15]

El objetivo fue crear un sistema operativo que pudiera competir en su momento con el de Palm o el Windows Phone de Microsoft y actualmente además con Android de Google Inc., iOS de Apple Inc. y BlackBerry OS.

Entre sus características están: [16]

- **Rendimiento:** Diseñado para optimizar el tiempo de vida de la batería.
- **Gestión de memoria optimizada:** para sistemas empujados (ejecutables de pequeño tamaño y códigos basados en ROM).
- **Reducidos requerimientos de memoria** en tiempo de ejecución.
- Las aplicaciones están orientadas al **manejo de eventos** en lugar de tener múltiple hilos de ejecución.

- **Multithreading:** es posible, pero generalmente se evita su uso en las aplicaciones debido a que se generan varios kilobytes de sobrecarga por thread.

2.2.4 Blackberry OS

El **BlackBerry OS** es un sistema operativo móvil desarrollado por Research In Motion (RIM) para sus dispositivos BlackBerry. Dicho sistema operativo permite al usuario disfrutar de la multitarea y además, tiene soporte para diferentes métodos de entrada adoptados por RIM para su uso en dispositivos móviles, particularmente la *trackwheel*, *trackball*, touchpad y pantallas táctiles. [17]

“Su desarrollo se remonta la aparición de los primeros handheld en 1999. Estos dispositivos permiten el acceso a correo electrónico, navegación web y sincronización con programas como Microsoft Exchange o Lotus Notes aparte de poder hacer las funciones usuales de un teléfono móvil.” (Juan Prince. 2010, párr. 2) [17]

El sistema operativo de BlackBerry está claramente orientado a su uso profesional como gestor de correo electrónico y agenda.

Como en el SO Symbian, los desarrolladores pueden crear programas para BlackBerry pero en el caso de querer tener acceso a ciertas funcionalidades restringidas necesitan ser firmados digitalmente para poder ser asociados a una cuenta de desarrollador de RIM. [17]

2.2.5 Windows Phone

Windows Phone es un sistema operativo móvil desarrollado por Microsoft, que ha sido diseñado para Smartphone y otros dispositivos móviles. Forma parte de los sistemas operativos que cuentan con una interfaz de usuario natural. Basado en el núcleo del Windows CE, está diseñado para ser estéticamente similar a las versiones de escritorio de Windows. [18]

Un año después de su primera versión, acaban de lanzar al mercado su primera actualización, Windows Phone 7, que tiene como principales atractivos: [19]

- **Una página de inicio basada en paneles:** La interfaz da libertad para escoger, cambiar quitar o agregar los paneles que queramos.
- **Microsoft Office Mobile:** Word Mobile, Excel Mobile, PowerPoint Mobile y OneNote Mobile que son versiones de las aplicaciones Office adaptadas a un teléfono móvil.
- **MarketPlace:** se puedan probar las aplicaciones antes de comprarlas, algo que no tienen las demás plataformas móviles.

2.3 Java

Las aplicaciones de **Android** se han desarrollado utilizando el lenguaje **Java**. **Java** es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems (propiedad de Oracle). Este lenguaje fue desarrollado siguiendo la sintaxis de C y C++ pero desechando rasgos que inducen a errores como pueden ser los punteros, la herencia multiple o la asignación de memoria y además, a diferencia de C++, fue creado desde el principio para ser completamente orientado a objetos. Todo en **Java** es un objeto y reside en alguna clase. [20]

El lenguaje **Java** se creó con cinco objetivos principales: [20]

1. Debería usar la metodología de la programación orientada a objetos.
2. Debería permitir la ejecución de un mismo programa en múltiples sistemas operativos.
3. Debería incluir por defecto soporte para trabajo en red.
4. Debería diseñarse para ejecutar código en sistemas remotos de forma segura.
5. Debería ser fácil de usar y tomar lo mejor de otros lenguajes orientados a objetos, como C++.

Por esto, sus pilares básicos son tres: [20]

- **Orientado a objetos (OO):** se refiere a un método de programación y al diseño del lenguaje. Es una forma de programar más cercana a una interpretación de cómo lo haríamos en la vida real. La idea es diseñar el software de manera que los tipos de datos estén asociados a sus operaciones. Así pues, los datos y las operaciones se combinan para formar los objetos. Un objeto será una entidad que contiene unas propiedades (datos) y unas funciones (el código) que es capaz de realizar. El principio es separar lo que cambia de lo que permanece inalterable para así ofrecer una base más estable para el diseño. El principal objetivo es facilitar la gestión y el manejo de grandes proyectos, para así mejorar la calidad y reducir los proyectos fallidos.
- **Independencia de la plataforma:** Cualquier programa en el lenguaje **Java** pueden ejecutarse en cualquier tipo de hardware, es decir, *"write once, run everywhere"*. Para esto, el código fuente escrito en **Java** y se compila creando un archivo llamado "bytecode" que son interpretados por una máquina virtual Java (**JVM**), un programa escrito en código nativo de la plataforma destino, que interpreta y ejecuta el código.

Java es fácil de aprender por una gran variedad de razones. Existen infinidad de recursos para ayudar a aprender el lenguaje, incluyendo sitios web, tutoriales o libros. Además, **Java** es uno de los lenguajes de programación más enseñado y usado en el

planeta. Se utiliza para muchos tipos diferentes de proyectos de programación, sin importar su escala.

2.4 XML

XML (eXtensible Markup Language) es un metalenguaje cuya labor es definir unas reglas que deberá cumplir un lenguaje de marcas, es decir, ordena, estructura y describe un lenguaje de marcas. Por lo tanto no se trata de un lenguaje en particular, sino una manera de definir lenguajes. Fue desarrollado por el World Wide Web Consortium (W3C) y se trata de una versión abreviada del SGML (Standard Generalized Markup Language). [21]

Un documento **XML** es, simplemente, un documento de texto que ofrece una representación estructural de los datos. Consta de dos estructuras, una física y otra lógica. Físicamente, el documento **XML** está compuesto por entidades (una o más) con contenido e identificadas por un nombre. Lógicamente, contiene elementos (uno o más) delimitados por marcas de principio y fin. Cada uno de los elementos tienen un identificador genérico, y dentro de este, puede contener una serie de atributos (con un nombre y un valor). [22]

Los archivos **XML** utilizados en **Android** para elementos como layouts que usará la aplicación para un funcionamiento correcto. Creamos entonces la interfaz de usuario de la aplicación, una apariencia física. El uso de este tipo de archivos no es imprescindible en la programación para **Android** puesto que puedes usar código **Java** para configurar la apariencia, cambiar formatos, mostrar elementos, ocultarlos. El porqué de su uso radica en mantener un diseño limpio y separar el código de la funcionalidad propia de la aplicación del código de diseño, para así facilitar las modificaciones que se realicen. [23]

2.5 Justificación

Para la elección de la plataforma que más se ajuste a nuestras necesidades e intereses, hemos tenido en cuenta una serie de factores, siendo estos los que se presentan a continuación:

- **Cuota de mercado mundial y en España.**
- **Precio para desarrolladores.**
- **Número de Apps.**
- **Expectativas.**

Cuota de mercado mundial y en España:

A continuación se muestran dos gráficos con los datos obtenidos por StatCounter de la cuota de mercado de los sistemas operativos móviles. En el primero de ellos, se muestra la cuota de sistemas operativos para dispositivos móviles en España. En el segundo, la cuota en todo el mundo.

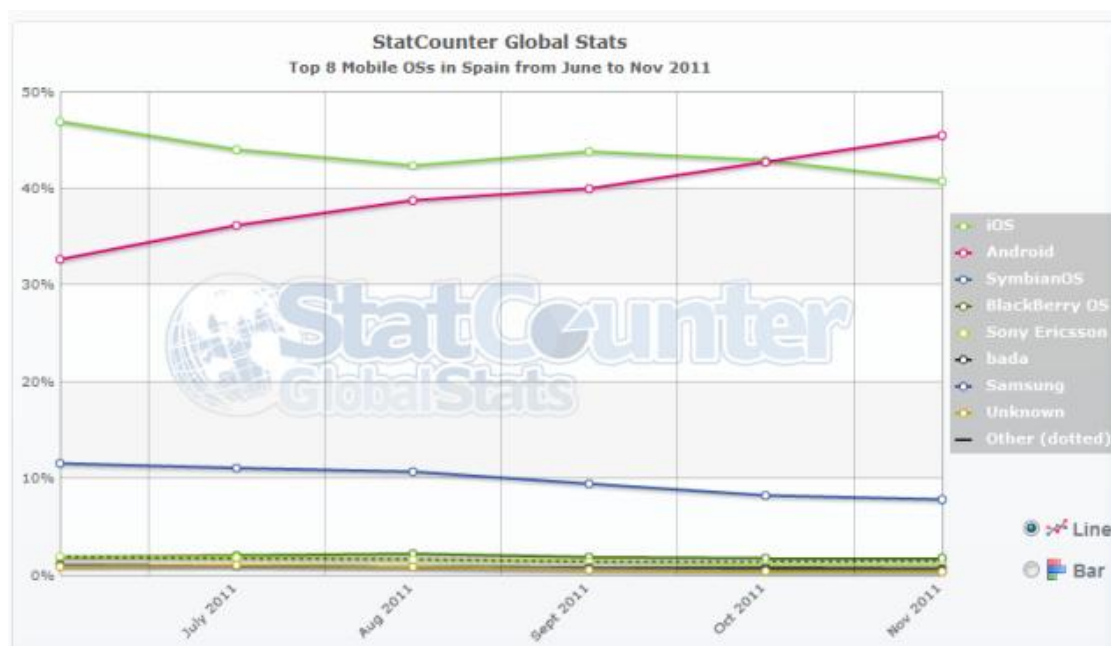


Figura 2. Cuota de sistemas operativos para dispositivos móviles en España [12]

Como se muestra en la gráfica, que cubre de Junio a Noviembre de 2011, a partir del mes de Octubre **Android supera a iOS** en cuota de mercado en España.

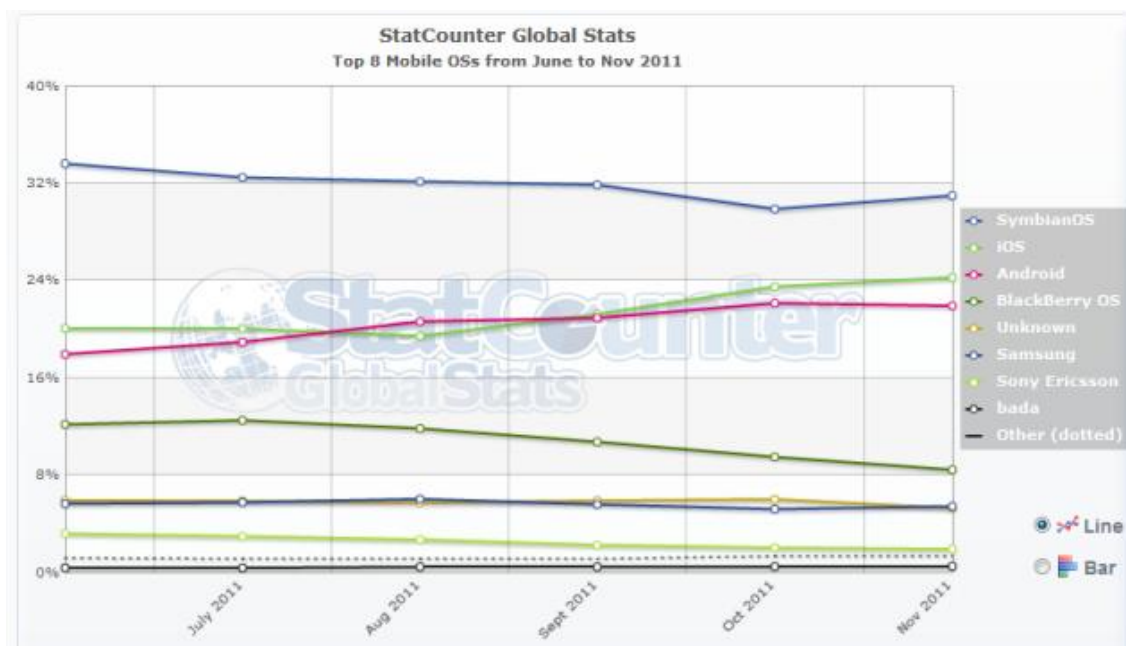


Figura 3. Cuota de sistemas operativos para dispositivos móviles en el mundo [12]

En cuanto al mercado mundial, Symbian es el líder con una diferencia sustancial, seguido de iOS y de Android siguiéndolos muy de cerca. BlackBerry es uno de los peores situados con un 10% de la cuota, muy lejos aún de los tres sistemas líderes.

Estos datos dejan ver el gran crecimiento que está teniendo Android en este último año, al igual que iOS. Ambos están ganando terreno a Symbian.

Precio para desarrolladores:

Uno de los factores más importantes es el precio que deben pagar para todas aquellas personas que decidan desarrollar aplicaciones para alguna de las plataformas móviles más importantes. Los precios que se muestran a continuación se refieren a licencia por año. Es decir, se deben renovar anualmente:

- **Android** → 25 \$
- **IOS** → 99 \$
- **BlackBerry** → 200 \$ (como máximo)
- **Windows Phone** → más de 300 €
- **Symbian OS** → 200 \$

Android se muestra como una de las plataformas más baratas para programadores jóvenes o con pocos recursos que quieran subir sus aplicaciones al **market**. Además de este hecho, **Android** utiliza código abierto, y con su kit de desarrollo no tendríamos que pagar ningún tipo de licencia para crear un **app**.

Número de Apps:

La compañía holandesa Distimo, firma holandesa que hace mediciones del mercado de telefonía móvil, ha concluido un estudio sobre el número de aplicaciones disponibles en los market más importantes. En la siguiente gráfica se muestran en color verde las aplicaciones gratuitas y en color rosa las aplicaciones de pago. [24]

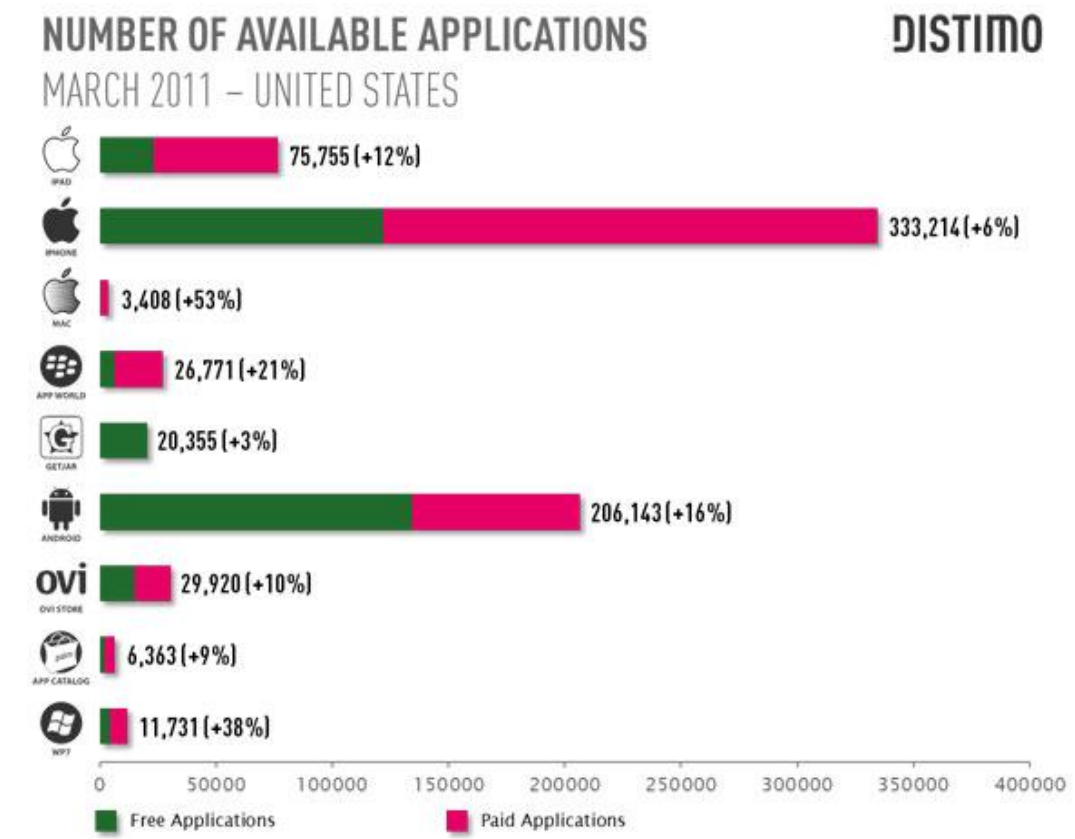


Figura 4. Número Total de Aplicaciones en losMarkets[24]

El Market de **Android**, con 134.342 aplicaciones gratuitas, supera las 121.845 de la App Store de **Apple**. Además, el informe revela que Android tiene un 65% de aplicaciones gratuitas por un 63% de **Apple**, lo que supone que Apple está recibiendo ingresos superiores a los que recibe Google en concepto de ventas. [24]

En la siguiente gráfica, proporcionada también por la compañía Distimo, se muestran los precios medios de cada app store. Más de la mitad de las aplicaciones disponibles en todas las tiendas tienen un precio máximo de entorno 2 euros o menos, excepto el markert de Blackberry y Windows Mobile que ofertan aplicaciones a menos coste. Clasificando las tiendas de menor a mayor coste, quedarían como más baratas Palm, Nokia y Android, siendo Apple Windows y Blackberry las más caras. [25]

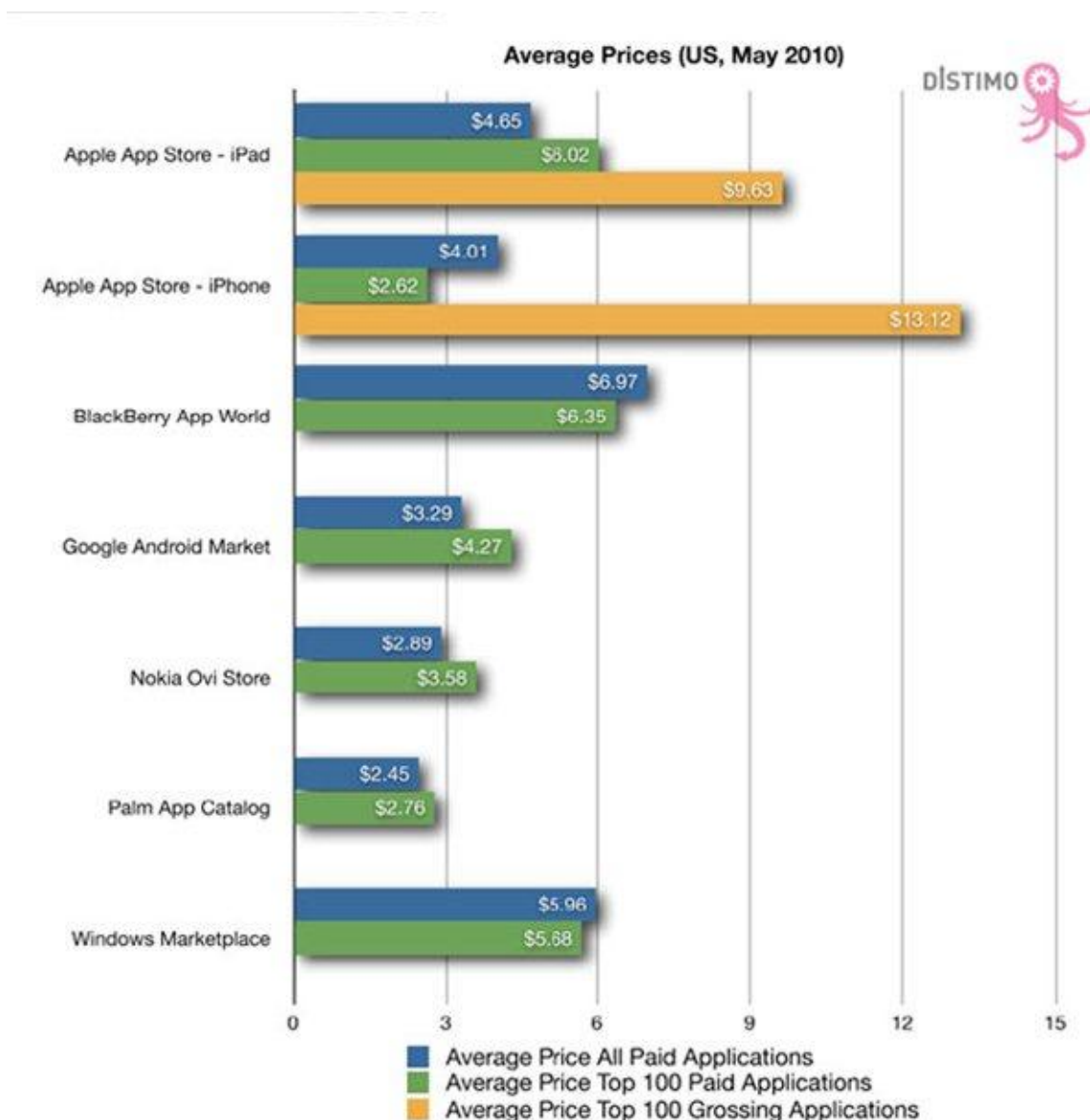


Figura 5. Precio Medio de las Apps [25]

Expectativas:

Según un informe de la compañía **Gartner**, **Android** sería el sistema operativo más usado alcanzando un 50% de mercado a finales del año 2012. El mercado de los Smartphone y de sus sistemas operativos es un mundo en constante cambio, evolucionando cada día, en el que no te puedes quedar dormido, porque en unos meses puede cambiar drásticamente la situación de privilegio de una compañía. [26]

Un ejemplo de ello es la marca **Nokia**, que cambiaron, después de mucho tiempo con **Symbian**, los sistemas operativos de sus dispositivos móviles de gama alta por **Windows Phone**. Y según el informe de **Gartner**, se colocará como la tercera plataforma más usada en 2012. [26]

Android permanecerá, con la mitad de la cuota de mercado, como sólido líder, mientras que **iOS** baja al tercer puesto. Malas noticias para **RIM** y sobre todo para **Symbian**

puesto que, aunque Blackberry continúa con una cuota de mercado similar, Symbian queda con un 0,1%. [26]

Una vez expuestos los anteriores argumentos, podemos decir que hay dos plataformas que sobresalen con respecto a las demás, **Android de Google** e **IOS de Apple**.

Se ha escogido el sistema operativo para dispositivos móviles desarrollado por **Google: Android**. Además de lo visto con anterioridad como **SO** con mayor crecimiento, mayor número de **apps** gratuitas y el precio más bajo de licencia de desarrollador, se han tenido en cuenta los siguientes aspectos positivos:

- **basada en código fuente abierto**, por lo que el soporte es global, con un montón de programadores en todo el mundo dispuestos a corregir errores.
- **Las aplicaciones se desarrollan en Java**, un lenguaje maduro y ampliamente adoptado.
- Tiene un **SDK multiplataforma** que funciona en **Windows, Linux y Mac**, mientras que para crear aplicaciones **IOS** te hace falta un ordenador marca **Apple**.
- **Multitud de terminales** entre los que podemos elegir para usar este sistema operativo. Mientras que en el **iPhone**, sólo podemos usar los 3 teléfonos que **Apple** ha lanzado en los últimos 3 años, en **Android** podemos elegir entre más de 50 terminales de diversas marcas.
- La mayoría de los dispositivos **Android** pueden soportar más capacidad de memoria con **tarjetas SD**. Los dispositivos Apple no tienen esta capacidad.

Capítulo 3

LA PLATAFORMA ANDROID

3.1 Características

Android “es una pila de software para dispositivos móviles que incluye un sistema operativo, middleware y aplicaciones clave.” (Android Developers, Feb 2012). Sus principales características son: [27]

- **El Marco de la aplicación** permite la reutilización y sustitución de componentes.
- **Su máquina virtual Dalvik** está optimizada para dispositivos móviles.
- **Navegador integrado** basado en el motor de código abierto WebKit.
- **Gráficos optimizados** impulsados por una biblioteca de gráficos 2D; gráficos 3D basados en la especificación OpenGL ES 1.0
- **SQLite** para el almacenamiento de datos estructurados.
- **Medios de apoyo** para el audio común, videos y formatos de imagen.
- **Telefonía GSM.**
- **Bluetooth, EDGE, 3G y WiFi.**
- **Cámara, GPS, brújula y acelerómetro.**
- **Entorno de desarrollo completo** que incluye un emulador de dispositivos, herramientas para la depuración, la memoria y perfiles de rendimiento, y un plugin para el IDE de Eclipse.

3.2 Arquitectura

La arquitectura de **Android**, está formada por 5 componentes: [27]

- **Applications:** conjunto de aplicaciones básicas creadas usando el lenguaje de programación Java.
- **Application Framework:** Al ser una plataforma de código abierto, los desarrolladores tienen acceso total al código fuente de las aplicaciones, ofreciendo la capacidad de crear aplicaciones muy ricas e innovadoras. Además se evita que se generen grandes cantidades de componentes de aplicaciones distintos que responden a la misma acción.

- **Libraries:** Android incluye un conjunto de librerías C/C++ utilizadas por los diversos componentes del sistema Android.
- **Android RunTime:** Android incorpora bibliotecas que proporciona la mayor parte de la funcionalidad disponible en las bibliotecas base del lenguaje de programación Java.
- **Linux Kernel:** Android se basa en la versión 2.6 de Linux para los servicios como seguridad, gestión de memoria, gestión de procesos, pila de red, y el modelo de controlador.

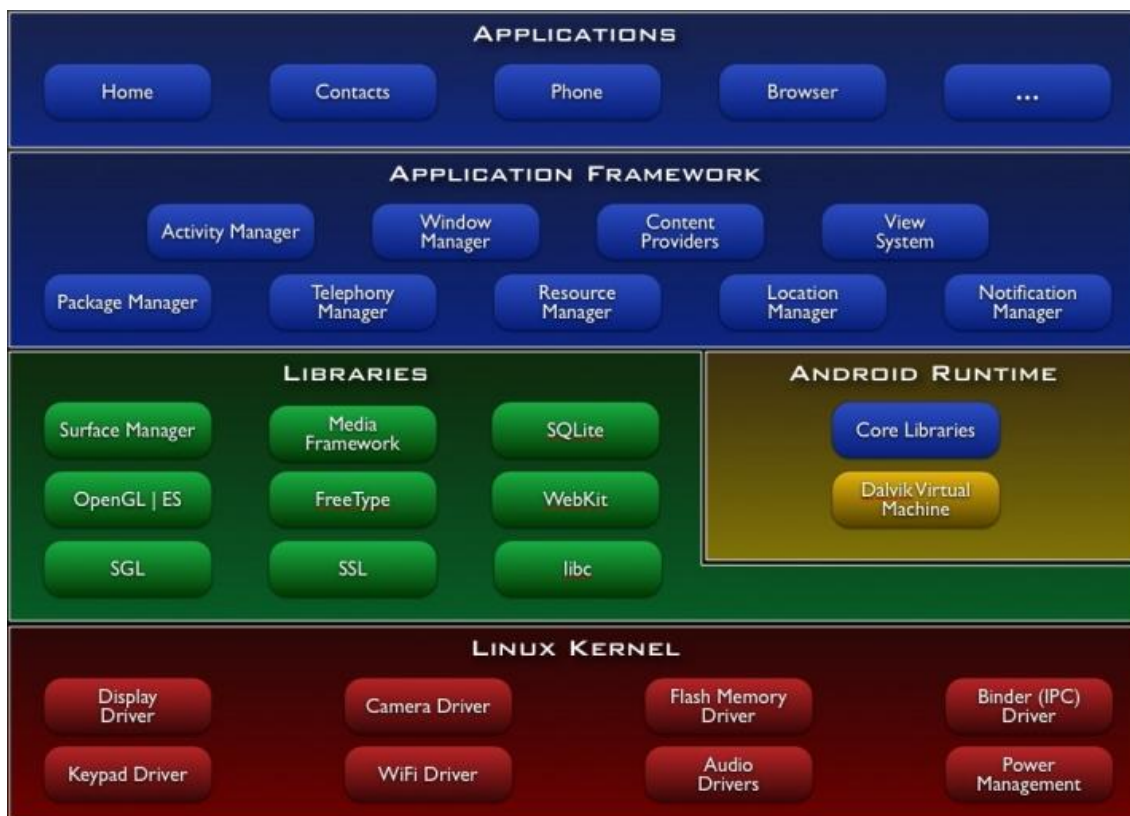


Figura 6. Arquitectura Interna de Android [27]

3.3 La Máquina Virtual Dalvik

Dalvik es una máquina virtual que ejecuta archivos en formato dex (Dalvik Executable), un formato optimizado para el almacenamiento eficiente y ejecución mapeable en memoria. Su objetivo fundamental, compilar el código a un bytecode independiente, e interpretándola la hora de ejecutar el programa.

Una de las características más importantes es que ha sido optimizada para que múltiples instancias de la máquina puedan funcionar al mismo tiempo con un coste en el rendimiento de la memoria muy bajo. Así, se protege a las aplicaciones abiertas ante un fallo o cierre inesperado de alguna de ellas.

3.4 Instalación del Entorno y Kit de Desarrollo.

Comenzaremos instalando el entorno de desarrollo, en este caso Eclipse, y posteriormente el kit de desarrollo de Android SDK.

Para la descarga del entorno de desarrollo eclipse iremos a su sitio web (<http://www.eclipse.org/downloads/packages/eclipse-ide-java-developers/heliossr2>) clicaremos sobre el archivo compatible con nuestro sistema operativo y una vez finalizada la descarga, lo descomprimos y arrancamos puesto que no requiere instalación.

Para la instalación del SDK de Android, iremos a su sitio web (<http://developer.android.com/sdk/index.html>) y descargaremos la versión compatible con nuestro sistema operativo. Instale el paquete SDK.

A continuación, instale el ADT plugin para Eclipse y añada las plataformas Android y otros componentes de su SDK.

3.5 Interfaz de Usuario

En Android, la interfaz de usuario se construye a partir de los tipos **View** y **ViewGroup**. El tipo **View** es la base para los llamados “**widgets**”, objetos totalmente implementados como campos de texto o botones. El tipo **ViewGroup** es la base para los **layout**, que ofrecen diferentes tipos de vista, como **framelayout**, **relativelayout**, **linearlayout**, **tablelayout**. [28]

Es decir, colocamos una serie de objetos view (Texto, imágenes, botones...) dentro de un viewGroup que nos lo colocará según el tipo de vista que tenga. La siguiente figura muestra un ejemplo de la estructura comentada.

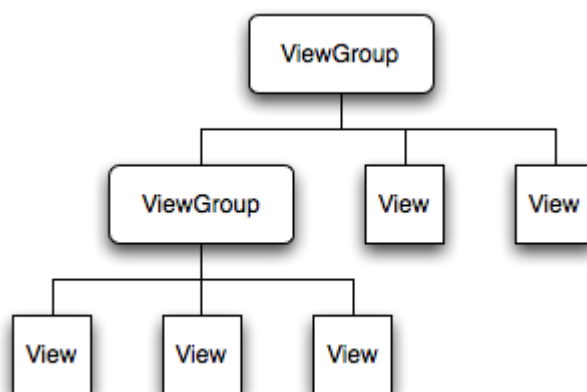


Figura 7. Jerarquía que define una interfaz de usuario [28]

La forma más común de establecer esta jerarquía es con un archivo XML que contenga la disposición de los distintos elementos. Por ejemplo un elemento `<TextView>` crea un objeto `TextView` en la interfaz, y un elemento `<LinearLayout>` crea un objeto `ViewGroup` equivalente. Por ejemplo, el siguiente fragmento de código crearía un `ViewGroup` en el que sus elementos, un cuadro de texto y un botón, se mostrarán uno a continuación del otro:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <TextView android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a TextView" />
    <Button android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a Button" />
</LinearLayout>
```

Código 01. Ejemplo simple con texto y un botón.

3.6 Estructura de un proyecto Android en Eclipse

Al crear un proyecto nuevo con el entorno de desarrollo eclipse, se generan una serie de carpetas para posteriormente generar la aplicación. La estructura para el proyecto llamado `HolaMundo` es la siguiente: [29]

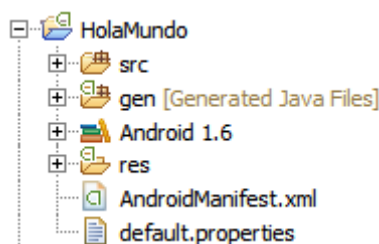


Figura 8. Estructura de un proyecto Android

A continuación pasaremos a explicar el contenido de cada carpeta:

- **Carpeta /src/:** Contiene todo el código fuente de la aplicación. Aquí es donde debe agregarse todo el código java.
- **Carpeta /res/:** Contiene todos los ficheros de recursos necesarios para la aplicación: imágenes, audios, cadenas de texto...

- **Res/drawable/**: Contienen las imágenes de la aplicación dependiendo de la resolución de la misma.
- **Res/layout/**: Contienen los archivos XML de definición de las diferentes pantallas de la interfaz gráfica.
- **Res/values/**: Contiene otros recursos de la aplicación como por ejemplo cadenas de texto (strings.xml), estilos (styles.xml), colores (colors.xml), etc.

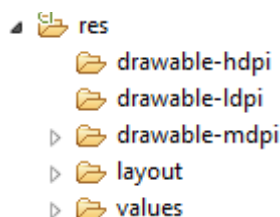


Figura 9. Estructura Carpeta resc de un proyecto Android

- **Carpeta /gen/**: Contiene una serie de elementos de código generados automáticamente al compilar el proyecto.
- **AndroidManifest**: Contiene la definición en XML de los aspectos principales de la aplicación, como su nombre, versión, icono, pantallas, mensajes...o los permisos necesarios para su ejecución.

3.7 Primer Arranque del SDK de Android

Para abrir o crear un Virtual Device tenemos dos posibilidades:

- 1) Vamos a la carpeta de nuestro **SDK** de Android, en **tools**, y arrancamos el ejecutable **android.exe**.
- 2) Abrimos **Eclipse** y seleccionamos el menú "**Window->Android SDK and AVD Manager**"

La primera vez que arranca, no se tiene aun ninguna versión de Android descargada, por lo que aparece una ventana para su descarga. Es recomendable bajar todas las versiones aunque solo se quiera programar para una de ellas.

Una vez instaladas, creamos el nuevo virtual device. Para ello, pulsamos la pestaña **Virtual Devices** y le damos al botón New.

- Introducimos el nombre.
- En **Target** seleccionamos el sistema operativo que va a tener nuestro dispositivo.

- Si las aplicaciones van a utilizar la tarjeta SD del teléfono, se puede marcar el tamaño a dar para utilizar en el dispositivo virtual.
- En **Skin** indicamos el tamaño de pantalla del teléfono.
- Y en **Hardware** le indicamos que funcionalidades va a tener nuestro dispositivo, tales como GPS, acelerómetros, etc. etc.
- Pulsamos el botón Create AVD y sólo nos queda abrir nuestro nuevo AVD.

3.8 Primera aplicación en Android: “Hola Mundo”

Para esta primera toma de contacto se ha elegido el programa HolaMundo desarrollado en la página paratuandroid por Lawprier por dos motivos:

1. Todo el código está explicado paso a paso y palabra a palabra por lo que es difícil perderse.
2. Modificamos tanto el archivo Java como el XML para empezar a familiarizarnos con su estructura.

Una vez creado el AVD ya tenemos preparado el entorno donde ejecutaremos nuestra primera aplicación, ahora es el momento de crear el proyecto que la contendrá. Desde el menú “**File > New > Other...**” de Eclipse o bien “**File > New > Project...**”, seleccionamos “**Android Project**”, dentro de la rama de “**Android**”. [30]

A continuación deberemos rellenar los siguientes campos:

- **Project Name:** El nombre que le vamos a dar a nuestro proyecto. En este caso HolaMundo.
- **Contents:** Elegiremos si vamos a crear un proyecto desde cero, o queremos usar algún tipo de código que ya tengamos escrito de antemano, y donde almacenarlo.
- **Build Target:** Seleccionaremos la versión de sistema operativo que vamos a utilizar.
- **Application name:** El nombre que va a tener nuestra aplicación.
- **Package name:** El código va a estar ordenado en paquetes, por convenio, suelen empezar por letra minúscula. Deberá de ser un nombre único. Google sugiere que como base se ponga: **nombreakaplicacion.nombredesarrollador**
- **Create Activity:** Crea la primera clase, donde escribiremos el código.

- **Min SDK version:** indicar el mínimo nivel de API requerido por la aplicación.

El Nivel de API es un valor entero que identifica de forma exclusiva la revisión de la API marco que ofrece una versión de la plataforma Android.

Pulsamos el botón “**finish**”. En este momento, eclipse nos crea toda la estructura necesaria de ficheros y directorios para nuestra aplicación.

Tras esto tendremos el proyecto preparado para comenzar a trabajar con él. Tenemos que abrir el archivo “HolaMundo.java” ubicado en la carpeta src si aún no lo hemos abierto. Iremos modificando su contenido poco a poco, explicando la funcionalidad de cada línea añadida. Cabe decir, que para crear cualquier aplicación, no hay una única solución correcta, sino que existe multitud de variantes según el programador que las realice. Esta es una de ellas, pero no es la única forma de hacerlo.

Para empezar a modificar el código, abrimos el archivo “HolaMundo.java” y añadimos unos atributos, que van a ser el texto que va a mostrar, y su color.

```
private String textoMostrar = "Hola Mundo!";
private int colortexto = Color.GREEN;
```

Código 02. Código java del archivo HolaMundo.java de Eclipse

La primera línea, palabra por palabra:

- **private:** determinar si los demás objetos, pueden o no acceder a la información que hay en la variable. En este caso es un atributo privado.
- **String:** tipo de dato del atributo, en este caso, una cadena de texto.
- **textoMostrar:** Nombre del atributo de tipo String, que va a contener el texto que vamos a querer mostrar, y que, cuando queramos referirnos a el durante el resto del programa, lo deberemos hacer por este nombre que le estamos dando.
- **“Hola Mundo!”:** Texto que vamos a mostrar, que va a quedar almacenado dentro del objeto textoMostrar.
- **“;”:** Fin de instrucción.

Y la segunda línea:

- **private:** Lo mismo que en la sentencia anterior.
- **int:** tipo de dato del atributo, en este caso, un entero.

- **colortexto**: Nombre del atributo de tipo int, que en este caso, va a almacenar el código de un color.
- **Color.GREEN**: Accedemos al atributo green de la clase color. Green contiene un entero con el código del color verde.

Una vez llegado a este punto, pasaremos a crear la interfaz que mostrará el contenido de la variable “textoMostrar”. Para ello nos debemos fijar en la última línea de código de la siguiente figura.

```
/**Called when the activity is first created. */
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
}
```

Código 03. Código java del archivo HolaMundo.java de Eclipse

La instrucción setContentView nos dice que cargará como interfaz el archivo con nombre “main” de la carpeta layout. Eclipse ya se ha encargado de generarlo automáticamente cuando creamos el proyecto.

Existen dos posibilidades, utilizar el editor gráfico, o cambiar el código **XML** a mano. Para adentrarnos un poco más en el desarrollo de las interfaces vamos a crear el código a mano. Para ello, en la parte inferior de la pantalla, hay una pestaña llamada main.xml, al lado de Graphical Layout. Clicamos, y vemos el siguiente código:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <TextView
        android:id="@+id/hello"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"></TextView>
</LinearLayout>
```

Código 04. Código xml del archivo Main.xml de Eclipse [30]

Este textview va a ser el objeto que utilizaremos para mostrar el texto.

Añadimos un identificador, para poder referirnos a él desde el código java. Lo haremos añadiendo la línea **android:text="@+id/cajatexto"**.

Volvemos al archivo con extensión java y añadimos las siguientes líneas de código dentro de la función onCreate para:

- Pedir la caja de texto: **TextView tv = (TextView) findViewById(R.id.cajatexto);**
Referenciamos el objeto con id cajatexto para trabajar con él.

- Fijar el texto: **Tv.setText(textoMostrar);** Le pasamos el texto guardado en la variable textoMostrar.
- Fijar el color: **Tv.setTextColor(colorTexto);** Le pasamos el color que guardamos anteriormente (verde).

Al introducir el código del objeto Textview, da un error en el margen izquierdo debido a que debemos importar código de otra clase, de la clase TextView. Botón derecho → Import TextView y queda subsanado el error.

Con esto finaliza el desarrollo del código. Para ejecutarlo, pulsamos el botón “Run” de la barra superior de iconos.

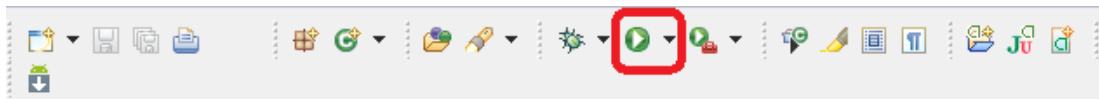


Figura 10. Barra superior de iconos de Eclipse

La primera vez que ejecutemos una aplicación, debemos decirle que se trata de una aplicación Android. Arranca el emulador del teléfono móvil android (AVD) y muestra el resultado.



Figura 11. Ventana del emulador SDK de Android [30]

Capítulo 4

ANÁLISIS

4.1 Introducción a Mobile Test Android Player

El objetivo básico de Mobile Test Android Player es mostrar al usuario con un formato u otro, dependiendo del tipo de material que desee ver, el material educativo seleccionado e interactuar con él para el desarrollo y corrección del mismo. Para ello utiliza fundamentalmente una **conexión a Internet**, con el fin de poder descargar el material educativo de su interés.

Mediante el establecimiento de una **conexión a Internet** bien a través de una tarifa de datos o bien a través de una conexión WIFI, la aplicación descarga del sitio web donde se encuentran los diversos materiales educativos (comprimidos en formato Zip) dichos paquetes, los descomprime y los almacena en la tarjeta de memoria del terminal (creando en ella una nueva carpeta donde serán guardados).

Cabe destacar, que no es imprescindible disponer de **conexión a Internet** para obtener dichos paquetes educativos. Para ello, con la ayuda de un ordenador, se pueden descargar en el mismo y mediante una **conexión USB** guardarlos en la carpeta correspondiente a la aplicación en el Smartphone.

Una vez que el usuario ha obtenido el material educativo de su interés, obtendrá toda la funcionalidad de la aplicación. Esta funcionalidad, una vez que el usuario elige el ejercicio que quiere realizar, le permitirá, además de ver el contenido seleccionado y navegar por él, conocer el número de respuestas acertadas, la solución correcta y repetir ejercicio.

4.2 Arquitectura General del Sistema

El siguiente diagrama muestra la arquitectura general del sistema, es decir, qué partes intervienen y cómo lo hacen.

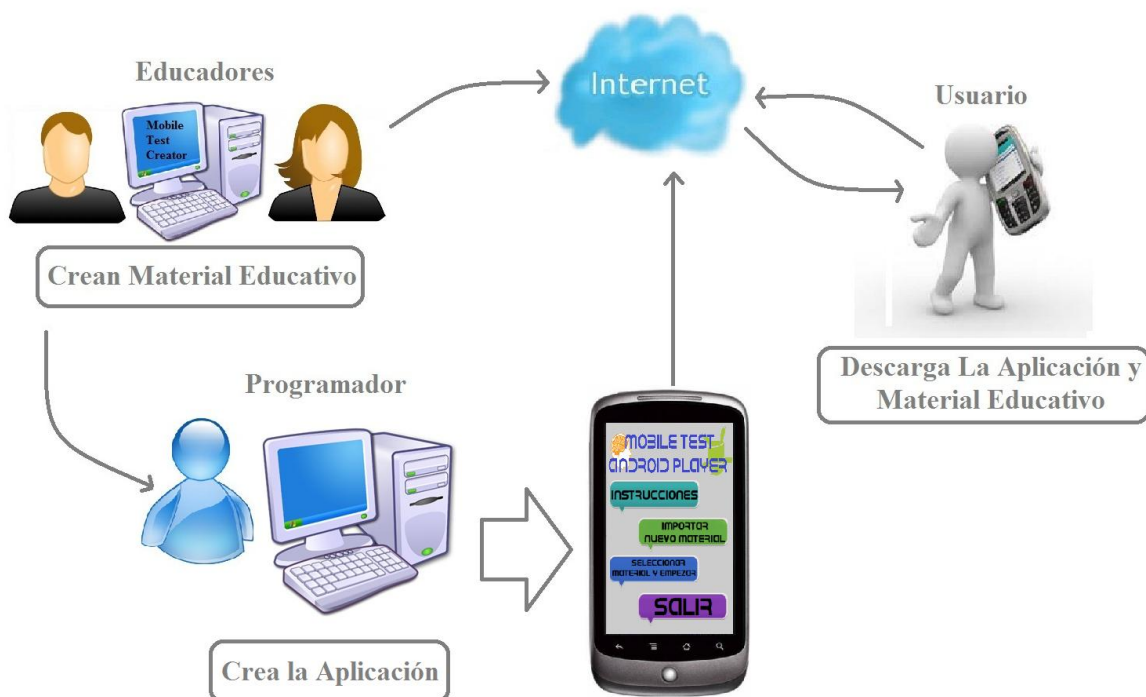


Figura 12. Diagrama General del Sistema

Se puede observar en la Figura la existencia de tres grupos de personas claramente diferenciados por su participación en el sistema:

- **Educadores:** Estos, con ayuda de la herramienta Mobile Test Creator (Anexo A), crean colecciones de material educativo con el contenido que deseen y las suben a la nube para que los usuarios se las descarguen.
- **Programador:** Con los materiales educativos facilitados por los educadores, crea una aplicación para mostrar su contenido y poder realizar los ejercicios. Esta aplicación se subirá a la nube para su descarga.

- Usuario: Accede a la nube para descargarse en su Smartphone tanto la aplicación como los materiales educativos de su interés.

Este proyecto se encarga de la parte realizada por el programador, realiza una aplicación genérica, que se pueda usar con cualquier material educativo creado por el grupo de educadores y que muestre toda la funcionalidad acordada.

4.3 Casos de Uso

Según el cuerpo docente de la asignatura “Diseño de Software Avanzado” del Departamento de Informática de Leganés de la Universidad Carlos III de Madrid, un caso de uso es una “forma de usar” el sistema, habitualmente descrita a través de un conjunto de “usos típicos”. [31]

Describe cómo un actor usa un sistema para conseguir un **objetivo**, y lo que el sistema hace para ayudarle. Cuenta la **historia** de cómo el sistema y sus actores colaboran para producir algo de valor, un uso **completo** del sistema. [32]

A lo largo de este apartado se especificarán los casos de uso para la aplicación que será desarrollada. Cada caso de uso se especificará mediante dos elementos:

- Diagramas para mostrar visualmente la interacción del usuario con el sistema.
- Tabla documento que describe el caso de uso.

La tabla explicativa tendrá el siguiente formato: [32]

- **Nombre:** identificador del caso de uso
- **Descripción:** describe los pasos realizados por el usuario para la situación planteada.
- **Actores:** especifica un **rol que adopta una entidad externa** que interacciona directamente con el sistema

- **Precondiciones:** condiciones que deben darse para la realización del caso de uso.
- **Flujo normal:** secuencia de acciones que describe la funcionalidad del caso de uso cuando no existen errores.
- **Flujo Alternativo:** secuencia de acciones que describe la funcionalidad del caso de uso después de ocurrir un error.
- **Postcondiciones:** condiciones que son resultado de la ejecución del caso de uso.

Nombre:	
Descripción:	
Actores:	
Pre-condiciones:	
Flujo Normal:	
Flujo Alternativo:	
Postcondiciones:	

Tabla 01. Tabla Casos de Uso

A continuación mostramos los casos de uso relacionados con nuestro sistema.



Figura 13. Diagrama de Casos de Uso

Nombre:	VER INSTRUCCIONES
Descripción:	Permite al usuario ver una serie de instrucciones básicas para el manejo de la aplicación.
Actores:	Usuario.
Pre-condiciones:	Tener instalada la aplicación en el Smartphone.
Flujo Normal:	<ol style="list-style-type: none"> 1. Seleccionar la opción “Instrucciones”. 2. Esperar la carga de la información. 3. Una vez que el usuario termine, pulsar el botón “atrás”.
Flujo Alternativo:	No Aplica.
Postcondiciones:	La aplicación muestra una nueva pantalla con la información de las instrucciones.

Tabla 02. Tabla Caso de Uso Ver Instrucciones

Nombre:	IMPORTAR NUEVO MATERIAL
Descripción:	Permite al usuario la descarga de un archivo comprimido ZIP desde una dirección web válida.
Actores:	Usuario
Pre-condiciones:	<ul style="list-style-type: none"> - Tener conexión a Internet o conexión Usb. - Tener instalada la aplicación en el Smartphone.
Flujo Normal:	<ol style="list-style-type: none"> 3. Seleccionar la opción “Importar Nuevo Material”. 4. Escribir la dirección Web de la descarga. 5. Pulsar el botón “Aceptar”.
Flujo Alternativo:	
Postcondiciones:	Material descargado, descomprimido y guardado en la carpeta de la aplicación en la tarjeta sd del Smartphone.

Tabla 03. Tabla Caso de Uso Importar Nuevo Material

Nombre:	SELECCIONAR MATERIAL
Descripción:	Permite al usuario seleccionar un material educativo para su posterior realización.
Actores:	Usuario.
Pre-condiciones:	<ul style="list-style-type: none"> - Tener la aplicación instalada en el Smartphone. - Tener descargado al menos una colección de material educativo.
Flujo Normal:	<ol style="list-style-type: none"> 1. Seleccionar la opción “Seleccionar Material Educativo” 2. Seleccionar una de las colecciones de materiales educativos previamente descargadas. 3. Seleccionar un test de la colección seleccionada.
Flujo Alternativo:	<ul style="list-style-type: none"> - No hay colecciones descargadas: Pulsar la tecla “atrás” del Smartphone para volver al menú inicial
Postcondiciones:	La aplicación muestra el primer ejercicio del test seleccionado.

Tabla 04. Tabla Caso de Uso Seleccionar Material Educativo

Nombre:	Salir de la aplicación
Descripción:	Permite al usuario cerrar la aplicación cuando desee.
Actores:	Usuario.
Pre-condiciones:	- Tener instalada la aplicación en el Smartphone.
Flujo Normal:	1. Seleccionar la opción “Salir”.
Flujo Alternativo:	
Postcondiciones:	Cierra la aplicación.

Tabla 05. Tabla Caso de Uso Salir

Nombre:	Acceder al Material – Navegar por el Test
Descripción:	Permite al usuario pasar de ejercicio en ejercicio dentro de un mismo test
Actores:	Usuario.
Pre-condiciones:	<ul style="list-style-type: none"> - Tener instalada la aplicación en el Smartphone. - Haber seleccionado un test de un material educativo - Que el Test tenga más de un ejercicio.
Flujo Normal:	<ol style="list-style-type: none"> 1. Pulsar la flecha orientada a la derecha para pasar al siguiente ejercicio, u orientada a izquierdas para volver al anterior.
Flujo Alternativo:	No Aplica
Postcondiciones:	Se muestra el ejercicio siguiente/anterior con toda su funcionalidad.

Tabla 06. Tabla Caso de Uso Navegar por el Test

Nombre:	Acceder al Material – Ver Anotaciones
Descripción:	Permite al usuario pulsando la pantalla ver el contenido completo de la anotación.
Actores:	Usuario.
Pre-condiciones:	<ul style="list-style-type: none"> - Tener instalada la aplicación en el Smartphone. - Haber seleccionado un test de un material educativo. - Encontrarse en un ejercicio de “Theory”. - Que el ejercicio tenga anotaciones que mostrar.
Flujo Normal:	<ol style="list-style-type: none"> 1. Pulsar sobre el texto en negrita y cursiva.
Flujo Alternativo:	<ol style="list-style-type: none"> 1. Pulsar sobre el texto en negrita y cursiva. 2. Si además de una anotación hay más texto que mostrar, pulsar ver anotación.
Postcondiciones:	Se muestra un pop up con el contenido de la anotación.

Tabla 07. Tabla Caso de Uso ver anotaciones

Nombre:	Acceder al Material – Consultar Puntuación
Descripción:	Permite al usuario ver la calificación obtenida de los ejercicios contestados del test.
Actores:	Usuario.
Pre-condiciones:	<ul style="list-style-type: none"> - Tener instalada la aplicación en el Smartphone. - Haber seleccionado un test de un material educativo. - Haber contestado al menos a uno de los ejercicios.
Flujo Normal:	1. Pulsar el botón “Show Score”
Flujo Alternativo:	<ul style="list-style-type: none"> - Mensaje de error al no haber contestado ningún ejercicio del test.
Postcondiciones:	Muestra en la pantalla el resultado obtenido.

Tabla 08. Tabla Caso de Uso consultar puntuación

Nombre:	Acceder al Material – Consultar Soluciones
Descripción:	Permite al usuario ver la solución correcta de los ejercicios contestados del test.
Actores:	Usuario.
Pre-condiciones:	<ul style="list-style-type: none"> - Tener instalada la aplicación en el Smartphone. - Haber seleccionado un test de un material educativo. - Haber contestado al menos a uno de los ejercicios.
Flujo Normal:	1. Pulsar el botón “Show Solution”
Flujo Alternativo:	<ul style="list-style-type: none"> - Mensaje de error al no haber contestado ningún ejercicio del test.
Postcondiciones:	Muestra en la pantalla las soluciones de los ejercicios contestados.

Tabla 09. Tabla Caso de Uso consultar soluciones

Nombre:	Acceder al Material – Corregir Test
Descripción:	Permite al usuario saber si la respuesta elegida es correcta o errónea.
Actores:	Usuario.
Pre-condiciones:	<ul style="list-style-type: none"> - Tener instalada la aplicación en el Smartphone. - Haber seleccionado un test de un material educativo. - Haber seleccionado una de las posibles respuestas del ejercicio a corregir.
Flujo Normal:	<ol style="list-style-type: none"> 1. Seleccionar una de las posibles respuestas del ejercicio. 2. Pulsar el botón “Check”.
Flujo Alternativo:	<ul style="list-style-type: none"> - Muestra un mensaje de error al no haber seleccionado una respuesta.
Postcondiciones:	Muestra en la pantalla un tic verde si la respuesta es correcta o un aspa roja si es incorrecta.

Tabla 10. Tabla Caso de Uso corregir test

Nombre:	Acceder al Material – Escuchar Audio
Descripción:	Permite al usuario escuchar el audio asociado al ejercicio.
Actores:	Usuario.
Pre-condiciones:	<ul style="list-style-type: none"> - Tener instalada la aplicación en el Smartphone. - Haber seleccionado un test de un material educativo. - Encontrarse en un ejercicio de “audio” o “audio choice”.
Flujo Normal:	1. Pulsar el botón “play audio”.
Flujo Alternativo:	No Aplica.
Postcondiciones:	Reproduce el sonido asociado al ejercicio.

Tabla 11. Tabla Caso de Uso Escuchar audio

Nombre:	Acceder al Material – Reiniciar Test
Descripción:	Permite al usuario volver a realizar el test borrando las respuestas, resultados y soluciones de los ejercicios contestados.
Actores:	Usuario.
Pre-condiciones:	<ul style="list-style-type: none"> - Tener instalada la aplicación en el Smartphone. - Haber seleccionado un test de un material educativo.
Flujo Normal:	1. Pulsar el botón “Reset”.
Flujo Alternativo:	No Aplica.
Postcondiciones:	Se reinicia el test y se muestra el primer ejercicio del mismo.

Tabla 12. Tabla Caso de Uso reiniciar test

Nombre:	Acceder al Material – Salir al Menú Inicial
Descripción:	Permite al usuario volver a la pantalla principal de la aplicación.
Actores:	Usuario.
Pre-condiciones:	<ul style="list-style-type: none"> - Tener instalada la aplicación en el Smartphone. - Haber seleccionado un test de un material educativo.
Flujo Normal:	1. Pulsar el botón “Exit”.
Flujo Alternativo:	No aplica.
Postcondiciones:	Vuelve al menú inicial de la aplicación.

Tabla 13. Tabla Caso de Uso salir al menú principal

4.4 Requisitos del Sistema

En este apartado llevaremos a cabo la definición, análisis y validación de los requisitos a partir de la información facilitada por el cliente. El objetivo es obtener un catálogo de los requisitos donde se pueda comprobar que el producto generado se ajusta a dichos requisitos.

Según la definición del IEEE [Piattini, 1996] el análisis de requisitos es el proceso del estudio de las necesidades de los usuarios desarrollado con el fin de obtener una definición de los requisitos del sistema, hardware o software, así como el proceso de estudio y refinamiento de dichos requisitos. Asimismo, se define requisito como una condición o capacidad que necesita el usuario para resolver un problema o conseguir un objetivo determinado [Piattini, 1996].

Los requisitos que vamos a registrar son:

- **Requisitos Funcionales:** Describen el funcionamiento del sistema. Definen los servicios que el sistema proporciona así como sus comportamientos a las diferentes entradas y situaciones.
- **Requisitos No Funcionales:** Describen las limitaciones del sistema, es decir, restricciones que afectan a sus funcionalidades, tales como rendimiento, interfaces, fiabilidad, seguridad, etc.

Según el estándar IEE 830, en su sección **3. Requisitos específicos**, lo ideal, aunque en la práctica no siempre realizable, es que los requisitos posean las siguientes características:

- **Corrección:** La ERS (Especificación de Requisitos Software) es correcta si y solo si todo requisito que figura aquí (y que será implementado en el sistema) refleja alguna necesidad real. La corrección de la ERS implica que el sistema implementado será el sistema deseado.
- **No ambiguos:** Cada requisito tiene una sola interpretación. Para eliminar la ambigüedad inherente a los requisitos expresados en lenguaje natural, se deberán utilizar gráficos o notaciones formales. En el caso de utilizar términos que, habitualmente, poseen más de una interpretación, se definirán con precisión en el glosario.
- **Completo:** Todos los requisitos relevantes han sido incluidos en la ERS. Conviene incluir todas las posibles respuestas del sistema a los datos de entrada, tanto válidos como no válidos.
- **Consistentes:** Los requisitos no pueden ser contradictorios. Un conjunto de requisitos contradictorio no es implementable.
- **Clasificados:** Normalmente, no todos los requisitos son igual de importantes. Los requisitos pueden clasificarse por grado de importancia (esencial,

condicional u opcional) o por estabilidad (cambios que se espera afecten al requisito). Esto sirve, ante todo, para no emplear excesivos recursos en implementar requisitos no esenciales.

- **Verificables:** la ERS es verificable si y solo si todos sus requisitos son verificables. Un requisito es verificable (testeable) si existe un proceso finito y no costoso para demostrar que el sistema cumple con el requisito. Un requisito ambiguo no es, en general, verificable. Expresiones como “a veces”, “bien”, “adecuado”, etc, introducen ambigüedad en los requisitos. Otros requisitos, como por ejemplo “en caso de accidente la nube toxica no se extenderá más allá de 25Km” no son verificable por el alto costo que conllevaría su verificación.
- **Modificable:** La ERS es modificable si y solo si se encuentra estructurada de forma que los cambios a los requisitos puedan realizarse de forma fácil, completa y consistente. La utilización de herramientas automáticas de gestión de requisitos (por ejemplo RequisitePro o Doors) facilitan enormemente esta tarea.
- **Trazables:** La ERS es trazable si se conoce el origen de cada requisito y se facilita la referencia de cada requisito a los componentes del diseño y de la implementación. La trazabilidad hacia atrás indica el origen (documento, persona, etc.) de cada requisito. La trazabilidad hacia delante de un requisito R indica qué componentes del sistema son los que realizan el requisito R.

Definiremos cada requisito según el siguiente estándar:

- **Identificación:** Código que identifica unívocamente el requisito. Debe respetarse el siguiente formato:

RQ-XXX-YYY-ZZ

Donde **XXX** es el identificador del tipo de requisito, siendo estos, funcionales (FUN) y no funcionales (NFN). **ZZ** es el número de secuencia que comienza, para cada tipo de requisito, en 01. **YYY** contempla los tipos de requisitos no funcionales y las siglas que se asociarán a los mismos serán:

SIGLAS	
INT	Requisitos de interfaz
USA	Requisitos de Usabilidad
REN	Requisitos de Rendimiento
OyA	Requisitos Operacionales y Ambientales
SOP	Requisitos de Soporte
SEG	Requisitos de Seguridad

Tabla 14. Tabla Siglas Requisitos

Los Requisitos No Funcionales se han obtenido de la Plantilla de especificación de requisitos Volere. [33]

- **Nombre:** Representa de forma breve y descriptiva el requisito, sin dar lugar a confusión con otros.
- **Prioridad:** Establecimiento de la importancia del requisito. La prioridad puede ser *alta*, *media* o *baja*.
- **Necesidad:** Establece el nivel de importancia del requisito en cuestión para el usuario. La necesidad puede ser *esencial*, *deseable* y *opcional*.
- **Claridad:** Evalúa el nivel en que un requisito es claro, evitando sobretodo la posibilidad de ambigüedad. La claridad puede tomar los valores *alta*, *media* y *baja*.
- **Verificabilidad:** Indica la medida en que se puede probar el software que da solución al requisito. Los posibles valores de la verificabilidad pueden ser *alta*, *media* y *baja*.
- **Estabilidad:** Establece el valor de la estabilidad del requisito. La estabilidad puede variar a lo largo de la vida del producto. Los posibles valores que puede tomar la estabilidad son *alta*, *media* y *baja*.
- **Descripción:** Descripción textual del requisito. Ésta debe ser breve pero intentando explicar de forma detallada la función del requisito.

La plantilla que usaremos será la siguiente:

Identificador: RQ-XXX-YY	
Nombre:	
Prioridad:	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad:	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad:	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción:	

Tabla 15. Plantilla de requisitos

4.4.1. Requisitos Funcionales

A continuación mostramos los requisitos funcionales de nuestro sistema:

Identificador: RQ-FUN-01	
Nombre:	Importar Material – Descarga de Material Educativo
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción:	Permitirá a los usuarios la descarga de material educativo en formato comprimido ZIP, introduciendo la dirección web donde se encuentra dicho material.

Tabla 16. Requisito Funcional 01

Identificador: RQ-FUN-02	
Nombre:	Importar Material – Descomprimir Material Educativo
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción:	Permitirá a los usuarios, una vez descargado el material educativo y de forma automática, la descompresión del archivo comprimido y su almacenamiento en la tarjeta de memoria del dispositivo móvil.

Tabla 17. Requisito Funcional 02

Identificador: RQ-FUN-03	
Nombre:	Instrucciones
Prioridad:	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad:	<input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja
Descripción:	Permitirá a los usuarios, visualizar una serie de instrucciones que a modo de manual de usuario, faciliten el aprendizaje del manejo de la aplicación.

Tabla 18. Requisito Funcional 03

Identificador: RQ-FUN-04	
Nombre:	Seleccionar Material – Listar Materiales Educativos
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción:	Permitirá a los usuarios, ver en forma de lista los nombres de los materiales educativos almacenados en la tarjeta de memoria.

Tabla 19. Requisito Funcional 04

Identificador: RQ-FUN-05	
Nombre:	Seleccionar Material – Seleccionar Material Educativo
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción:	Permitirá a los usuarios, seleccionar un material educativo almacenado en la tarjeta de memoria.

Tabla 20. Requisito Funcional 05

Identificador: RQ-FUN-06	
Nombre:	Seleccionar Material – Borrar Material Educativo
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción:	Permitirá a los usuarios, borrar los materiales educativos almacenados en la tarjeta de memoria.

Tabla 21. Requisito Funcional 06

Identificador: RQ-FUN-07	
Nombre:	Seleccionar Material – Listar Contenido Material Educativo
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción:	Permitirá a los usuarios ver en forma de lista los ejercicios, tanto prácticos como teóricos, que contiene un material educativo previamente seleccionado.

Tabla 22. Requisito Funcional 07

Identificador: RQ-FUN-08	
Nombre:	Seleccionar Material – Seleccionar Contenido Material Educativo
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción:	Permitirá a los usuarios, seleccionar un ejercicio del material educativo seleccionado.

Tabla 23. Requisito Funcional 08

Identificador: RQ-FUN-09	
Nombre:	Visualizar Ejercicio - Introduction
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción:	Permitirá a los usuarios, visualizar el contenido introductorio asociado a un ejercicio.

Tabla 24. Requisito Funcional 09

Identificador: RQ-FUN-10	
Nombre:	Visualizar Ejercicio - Theory
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción:	Permitirá a los usuarios, visualizar el contenido teórico asociado a un ejercicio.

Tabla 25. Requisito Funcional 10

Identificador: RQ-FUN-11	
Nombre:	Visualizar Ejercicio – Fill the Gap, Multiple Fill The Gap y Multiple Choice
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción:	Permitirá a los usuarios, visualizar el enunciado del ejercicio así como un máximo de cinco posibles respuestas.

Tabla 26. Requisito Funcional 11

Identificador: RQ-FUN-12	
Nombre:	Visualizar Ejercicio – Seleccionar Fill The Gap
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción:	Permitirá a los usuarios, seleccionar una de las posibles respuestas del ejercicio.

Tabla 27. Requisito Funcional 12

Identificador: RQ-FUN-13	
Nombre:	Visualizar Ejercicio – Word Order
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción:	Permitirá a los usuarios, visualizar el enunciado del ejercicio.

Tabla 28. Requisito Funcional 13

Identificador: RQ-FUN-14	
Nombre:	Visualizar Ejercicio – Ordenar Word Order
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción:	Permitirá a los usuarios, escribir una respuesta correspondiente al enunciado del ejercicio.

Tabla 29. Requisito Funcional 14

Identificador: RQ-FUN-15	
Nombre:	Visualizar Ejercicio – Reproducir Audio
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción:	Permitirá a los usuarios, reproducir un archivo de sonido asociado a los ejercicios del tipo Audio y Audio Choice.

Tabla 30. Requisito Funcional 15

Identificador: RQ-FUN-16	
Nombre:	Visualizar Ejercicio – Volumen Audio
Prioridad:	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción:	Permitirá a los usuarios, modificar el volumen del archivo de audio asociado a los ejercicios del tipo Audio y Audio Choice.

Tabla 31. Requisito Funcional 16

Identificador: RQ-FUN-17	
Nombre:	Visualizar Ejercicio – Audio Choice
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción:	Permitirá a los usuarios, visualizar de entre 1 a 8 grupos de preguntas cada una de ellas compuestas por 2 posibles respuestas.

Tabla 32. Requisito Funcional 17

Identificador: RQ-FUN-18	
Nombre:	Visualizar Ejercicio – Score
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción:	Permitirá a los usuarios, visualizar el número de preguntas acertadas del ejercicio realizado.

Tabla 33. Requisito Funcional 18

Identificador: RQ-FUN-19	
Nombre:	Visualizar Ejercicio – Solution
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción:	Permitirá a los usuarios, visualizar las respuestas correctas de los ejercicios realizados.

Tabla 34. Requisito Funcional 19

Identificador: RQ-FUN-20	
Nombre:	Visualizar Ejercicio – Reset
Prioridad:	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad:	<input type="checkbox"/> Esencial <input checked="" type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción:	Permitirá a los usuarios, reiniciar el ejercicio sin necesidad de salir al menú principal y seleccionarlo.

Tabla 35. Requisito Funcional 20

Identificador: RQ-FUN-21	
Nombre:	Visualizar Ejercicio – Exit
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción:	Permitirá a los usuarios, una vez terminado el ejercicio, volver al menú principal.

Tabla 36. Requisito Funcional 21

Identificador: RQ-FUN-22	
Nombre:	Visualizar Ejercicio – Corregir Ejercicio
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción:	Permitirá a los usuarios conocer si su respuesta es correcta o no mediante un marcador verde o rojo respectivamente. Esta funcionalidad está disponible en todos los ejercicios salvo en “introduction”, “theory” y “audio”.

Tabla 37. Requisito Funcional 22

Identificador: RQ-FUN-23	
Nombre:	Visualizar Ejercicio – Pasar Ejercicio
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción:	Permitirá a los usuarios, pasar de un ejercicio a otro hacia delante o hacia atrás (dentro de un mismo test), pudiendo dejar la pregunta sin responder para hacerlo posteriormente. Si se responde, se guardará la respuesta escogida y no dejará realizar otra vez dicho ejercicio

Tabla 38. Requisito Funcional 23

Identificador: RQ-FUN-24	
Nombre:	Visualizar Ejercicio – Pasar Gap
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad:	<input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción:	Permitirá a los usuarios, dentro del ejercicio Multiple Fill The Gap o Drag And Drop, pasar de un Gap a otro, sin necesidad de contestar. Si se responde, se guardará la respuesta escogida y no dejará realizar otra vez dicho ejercicio

Tabla 39. Requisito Funcional 24

Identificador: RQ-FUN-25	
Nombre:	Visualizar Ejercicio – Salir
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción:	Permitirá a los usuarios, salir de la aplicación.

Tabla 40. Requisito Funcional 25

4.4.2. Requisitos No Funcionales

Identificador: RQ-NFN-INT-01	
Nombre:	Apariencia
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción:	El producto debe tener un diseño capaz de atraer a jóvenes universitarios.

Tabla 41. Requisito de Interfaz 01

Identificador: RQ-NFN-INT-02	
Nombre:	Contenido
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción:	El producto debe tener pocas opciones que elegir.

Tabla 42. Requisito de Interfaz 02

Identificador: RQ-NFN-INT-03	
Nombre:	Logo
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción:	El producto debe tener un logo propio visible en la pantalla principal.

Tabla 43. Requisito de Interfaz 03

Identificador: RQ-NFN-INT-04	
Nombre:	Menú de desplazamiento
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción:	El producto debe tener un menú fijo en todas las pantallas del ejercicio, que permitirá desplazarse dentro de un ejercicio

Tabla 44. Requisito de Interfaz 04

Identificador: RQ-NFN-INT-05	
Nombre:	Apariencia Contenido
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción:	El producto debe ser capaz de visualizar el texto en formato html cuando sea necesario.

Tabla 45. Requisito de Interfaz 05

Identificador: RQ-NFN-INT-06	
Nombre:	Imágenes
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción:	El producto debe ser capaz de mostrar imágenes cuando sea necesario.

Tabla 46. Requisito de Interfaz 06

Identificador: RQ-NFN-INT-07	
Nombre:	Apariencia General
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción:	El producto debe tener una apariencia similar a su homólogo web.

Tabla 47. Requisito de Interfaz 07

Identificador: RQ-NFN-USA-01	
Nombre:	Facilidad de uso
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción:	El producto, al ser usado por personas que no recibirán entrenamiento antes, deberá ser fácil de usar.

Tabla 48. Requisito de Usabilidad 01

Identificador: RQ-NFN-USA-02	
Nombre:	Facilidad de aprendizaje
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción:	El producto será fácil de aprender por un estudiante universitario.

Tabla 49. Requisito de Usabilidad 02

Identificador: RQ-NFN-USA-03	
Nombre:	Idioma
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción:	La interfaz mostrará sus contenidos en inglés, mostrándose en español únicamente la interfaz principal.

Tabla 50. Requisito de Usabilidad 03

Identificador: RQ-NFN-REN-01	
Nombre:	Disponibilidad
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción:	El producto estará disponible para su uso las 24 horas del día, 7 días a la semana.

Tabla 51. Requisito de Rendimiento 01

Identificador: RQ-NFN-REN-02	
Nombre:	Descarga Nuevo Material
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción:	<p>El producto, para la descarga de un nuevo paquete de material educativo tiene 3 métodos:</p> <ol style="list-style-type: none"> 4. Conexión a Internet mediante una tarifa de datos. 5. Conexión a Internet mediante WIFI. 6. Conexión USB a un ordenador.

Tabla 52. Requisito de Rendimiento 02

Identificador: RQ-NFN-REN-03	
Nombre:	Conexión a Internet
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción:	El producto sólo hará uso de la conexión a internet en la descarga de un nuevo paquete de material educativo.

Tabla 53. Requisito de Rendimiento 03

Identificador: RQ-NFN-OyA-01	
Nombre:	Instalación
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción:	El producto podrá ser instalado por un usuario sin entrenamiento sin recurrir a instrucciones impresas por separado.

Tabla 54. Requisito Operacional y Ambiental 01

Identificador: RQ-NFN-SOP-01	
Nombre:	Plataforma
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción:	El producto será utilizado solamente en terminales móviles con sistema operativo Android.

Tabla 55. Requisito de Soporte 01

Identificador: RQ-NFN-SOP-02	
Nombre:	Versión
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción:	El dispositivo en el que se instale la aplicación debe contar con una versión de Android igual o superior a la 1.6.

Tabla 56. Requisito de Soporte 02

Identificador: RQ-NFN-SEG-01	
Nombre:	Autorización
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción:	Cualquier persona que se descargue la aplicación tiene acceso autorizado al producto.

Tabla 57. Requisito de Seguridad 01

Identificador: RQ-NFN-SEG-02	
Nombre:	Permisos
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Necesidad:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Deseable <input type="checkbox"/> Opcional
Claridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja
Descripción:	<p>Al instalar la aplicación el usuario debe otorgar los permisos necesarios a la aplicación para que esta pueda funcionar de forma correcta en el teléfono. Estos permisos son:</p> <ul style="list-style-type: none"> - Almacenamiento: modificar/eliminar contenido de la tarjeta SD. - Comunicación de red: acceso íntegro a Internet.

Tabla 58. Requisito de Seguridad 02

Capítulo 5

DISEÑO

5.1 Diagrama Detallado

Para diseñar un diagrama detallado y establecer los diferentes elementos que aparecerán en él, es necesario realizar una división a alto nivel de la aplicación. Ésta puede ser dividida en cuatro módulos, cada uno correspondiente a las principales funcionalidades que deberá llevar a cabo. Por un lado debe interactuar con el módulo de descarga para obtener los materiales educativos, y con el módulo de almacenamiento para guardarlos en la tarjeta de memoria del terminal móvil. Por otro lado deberá utilizar el módulo de lectura para leer el contenido del material educativo, para que así el módulo de despliegue pueda mostrarlo.

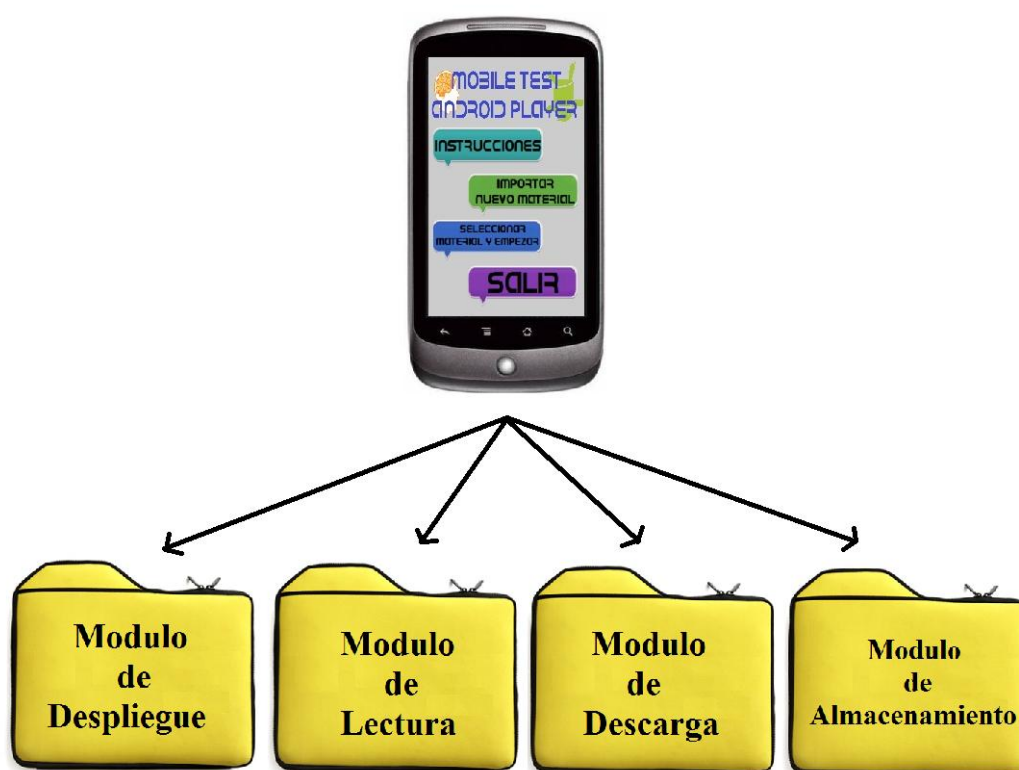


Figura 13. Diagrama Detallado de MobileTestAndroidPlayer

La siguiente figura se corresponde con el diagrama de componentes de la aplicación. En él se representan los principales componentes del sistema así como las relaciones entre ellos.

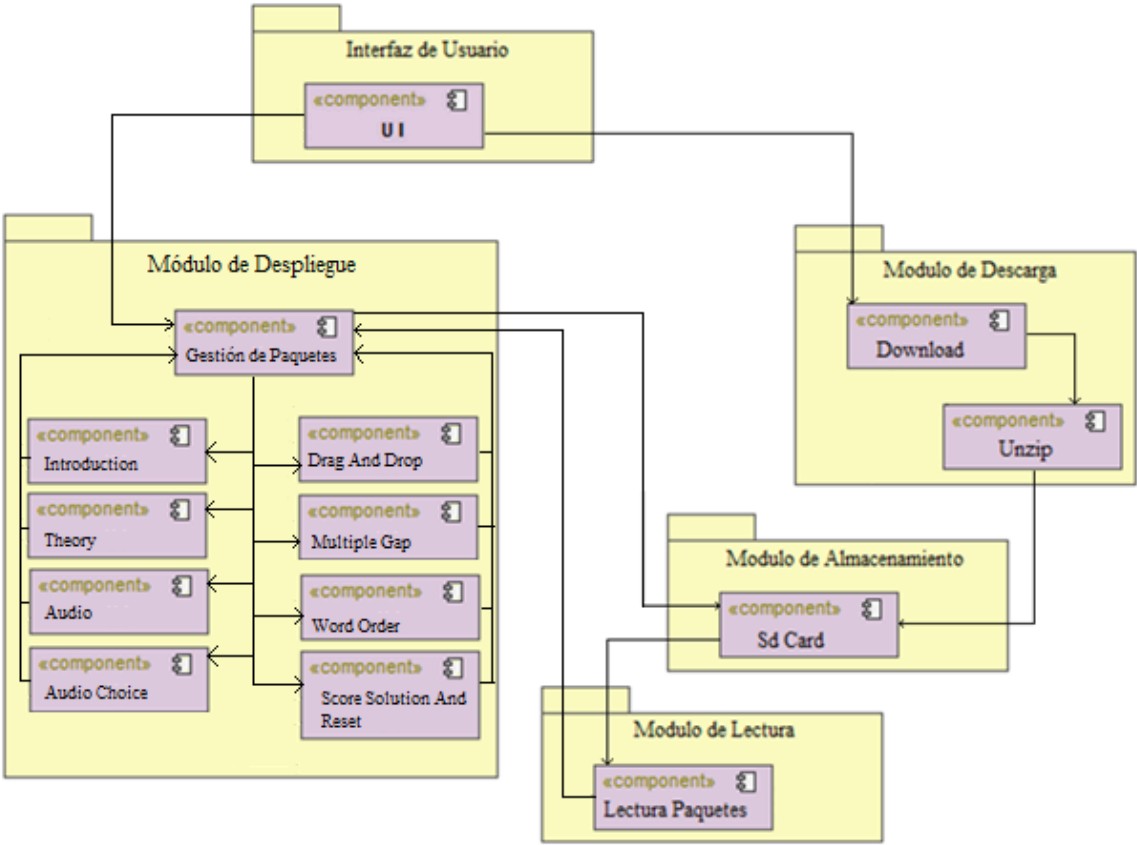


Figura 14. Diagrama de Componentes de MobileTestAndroidPlayer

Explicación de cada componente:

Tabla 59. Componente Interfaz de Usuario

Interfaz de Usuario	
Nombre:	Componente UI
Propósito:	Componente encargado de generar las pantallas. De forma más detallada, será el encargado de mostrar la pantalla principal de la aplicación.
Funciones:	<ul style="list-style-type: none">- Ver Instrucciones- Importar Material- Seleccionar Material- Salir

Módulo de Despliegue	
Nombre:	Componente gestión de Paquetes
Propósito:	Componente encargado de listar los materiales almacenados en el dispositivo de memoria, permite su navegación y selección, y llevar a cabo su carga.
Funciones:	<ul style="list-style-type: none"> - Listar - Seleccionar - Cargar - Leer Material

Tabla 60. Componente Gestión de Paquetes

Módulo de Despliegue	
Nombre:	Componente Introduction
Propósito:	Visualizar el contenido de la página. Dará la posibilidad al usuario de avanzar/retroceder al siguiente/anterior ejercicio del test.
Funciones:	<ul style="list-style-type: none"> - Visualizar - Anterior/Siguiente Ejercicio

Tabla 61. Componente Introduction

Módulo de Despliegue	
Nombre:	Componente Theory
Propósito:	Visualizar el contenido de la página, y dar soporte a toda su funcionalidad.
Funciones:	<ul style="list-style-type: none"> - Visualizar - Desplegar Contenido - Ver Anotación - Ver imagen - Anterior/Siguiente Ejercicio

Tabla 62. Componente Theory

Módulo de Despliegue	
Nombre:	Componente Audio
Propósito:	Visualizar el contenido de la página, y reproducir un archivo de audio asociado
Funciones:	<ul style="list-style-type: none"> - Visualizar - Reproducir Audio - Anterior/Siguiente Ejercicio

Tabla 63. Componente Audio

Módulo de Despliegue	
Nombre:	Componente Audio Choice
Propósito:	Visualizar el contenido de la página, reproducir un archivo de audio y dar soporte al usuario para contestar y corregir las preguntas que contenga.
Funciones:	<ul style="list-style-type: none"> - Visualizar - Reproducir Audio - Seleccionar Pregunta - Corregir Pregunta - Anterior/Siguiente Ejercicio

Tabla 64. Componente Audio Choice

Módulo de Despliegue	
Nombre:	Componente Word Order
Propósito:	Visualizar el contenido de la página, introducir texto y corregir el ejercicio.
Funciones:	<ul style="list-style-type: none"> - Visualizar - Introducir Texto - Corregir Pregunta - Anterior/Siguiente Ejercicio

Tabla 65. Componente Word Order

Módulo de Despliegue	
Nombre:	Componente Multiple Gap
Propósito:	Visualizar el contenido de las páginas Multiple fill the gap, Multiple Choice y Fill the Gap, seleccionar la respuesta y corregir el ejercicio. Además de poder avanzar/retroceder al siguiente/anterior ejercicio del test, podrá avanzar/retroceder dentro del ejercicio si este consta de más de una pregunta.
Funciones:	<ul style="list-style-type: none"> - Visualizar - Seleccionar Respuesta - Corregir Pregunta - Anterior/Siguiente Ejercicio - Anterior/Siguiente Pregunta

Tabla 66. Componente Multiple Gap

Módulo de Despliegue	
Nombre:	Componente Drag And Drop
Propósito:	Visualizar el contenido de la página, seleccionar la o las respuestas oportunas y corregir el ejercicio. Además de poder avanzar/retroceder al siguiente/anterior ejercicio del test, podrá avanzar/retroceder dentro del ejercicio si este consta de más de una pregunta.
Funciones:	<ul style="list-style-type: none"> - Visualizar - Seleccionar Respuesta/s - Corregir Pregunta - Anterior/Siguiente Ejercicio - Anterior/Siguiente Pregunta

Tabla 67. Componente Drag And Drop

Modulo de Descarga:

Módulo de Descarga	
Nombre:	Componente Download
Propósito:	Visualizar el contenido de la página e introducir una Url para la descarga de Material.
Funciones:	<ul style="list-style-type: none"> - Descarga desde Url.

Tabla 68. Componente Download

Módulo de Descarga	
Nombre:	Componente Unzip
Propósito:	Visualizar el contenido de la página e introducir y la descompresión del paquete descargado.
Funciones:	<ul style="list-style-type: none"> - Descomprimir

Tabla 69. Componente Unzip

Cabe destacar en este módulo, que para la descarga de material existe otra posibilidad no contemplada en el diagrama anterior puesto que no se trata de una tarea realizada por la aplicación sino por el propio usuario. Se trata de la descarga no mediante una conexión a Internet, sino a través de una conexión Usb.

El usuario deberá descargarse en un ordenador los materiales educativos que desee, y conectando el terminal móvil a este, descargarlo en la carpeta adecuada.

Módulo de Almacenamiento	
Nombre:	Componente Sd Card
Propósito:	Guardar en el dispositivo de almacenamiento del terminal móvil el material educativo descargado y descomprimido.
Funciones:	- Almacenar en Sd Card

Tabla 70. Componente Sd Card

Modulo de Almacenamiento:

El material educativo queda guardado en el directorio correspondiente siguiendo la estructura que se muestra a continuación:

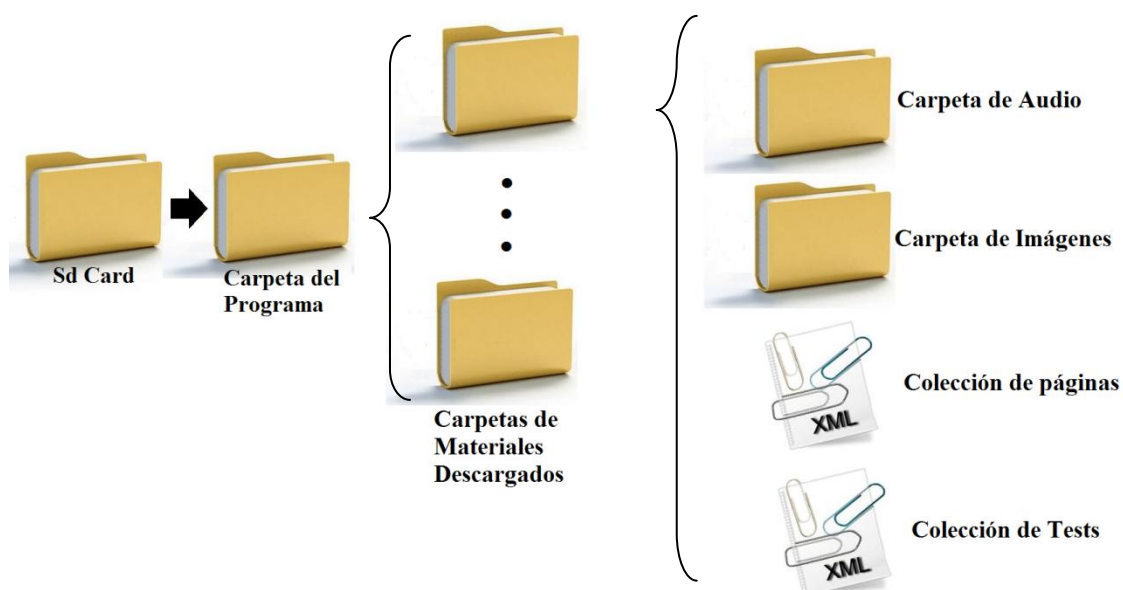


Figura 15. Directorio Almacenamiento Material Educativo

Todo material ya se descargado a través de una conexión a Internet o introducido en el dispositivo móvil mediante una conexión Usb, queda guardado en una carpeta con el nombre del material dentro de una carpeta con el nombre de la aplicación (donde estarán todos aquellos materiales que se hayan introducido). Todas estas carpetas se podrán encontrar en la tarjeta de memoria del dispositivo.

Módulo de Lectura:

Módulo de Lectura	
Nombre:	Componente Lectura de Paquetes
Propósito:	Leer los archivos xml del paquete seleccionado para posteriormente pasárselo al componente Gestión de Paquetes
Funciones:	- Leer

Tabla 71. Componente Lectura de Paquetes

5.2 Diagramas de Secuencia

Esta sección del presente documento nos detalla los diagramas de secuencia para mostrar qué objetos se comunican y qué mensajes generan esas comunicaciones. Cabe destacar, para un mejor entendimiento, que los diagramas se refieren a los componentes de los módulos y no a las clases de los componentes.

Existirá un diagrama por cada caso de uso. Estos son:

- Ver Instrucciones: Permite al usuario, pulsando el botón “instrucciones”, ver una serie de instrucciones básicas para el manejo de la aplicación.

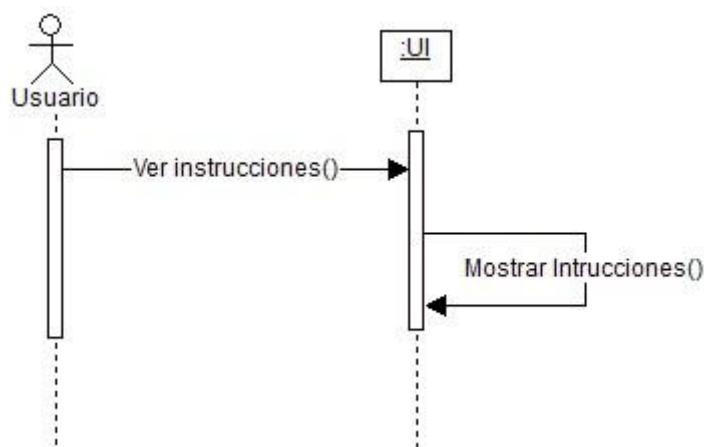


Figura 16. Diagrama de Secuencia Ver Instrucciones

- Importar Nuevo Material: Permite al usuario, pulsando el botón “importar nuevo material”, la descarga de un archivo comprimido ZIP desde una dirección web válida.

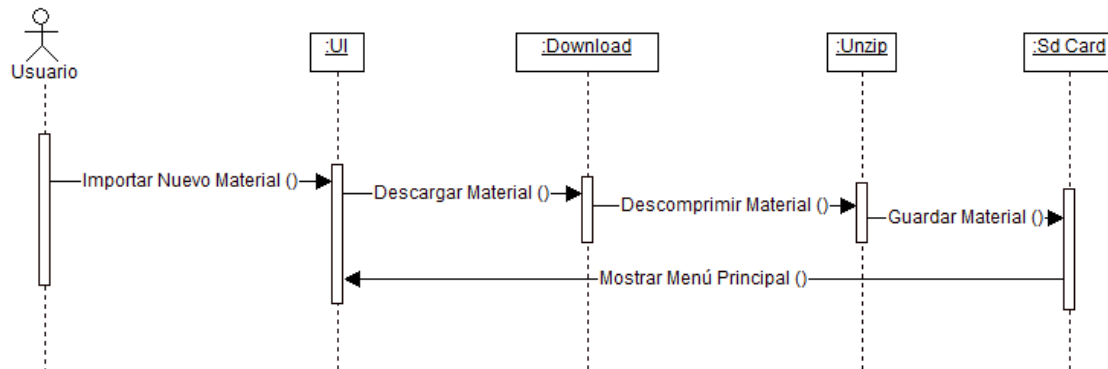


Figura 17. Diagrama de Secuencia Importar Nuevo Material

- Seleccionar Material: Permite al usuario seleccionar un material educativo para su posterior realización.

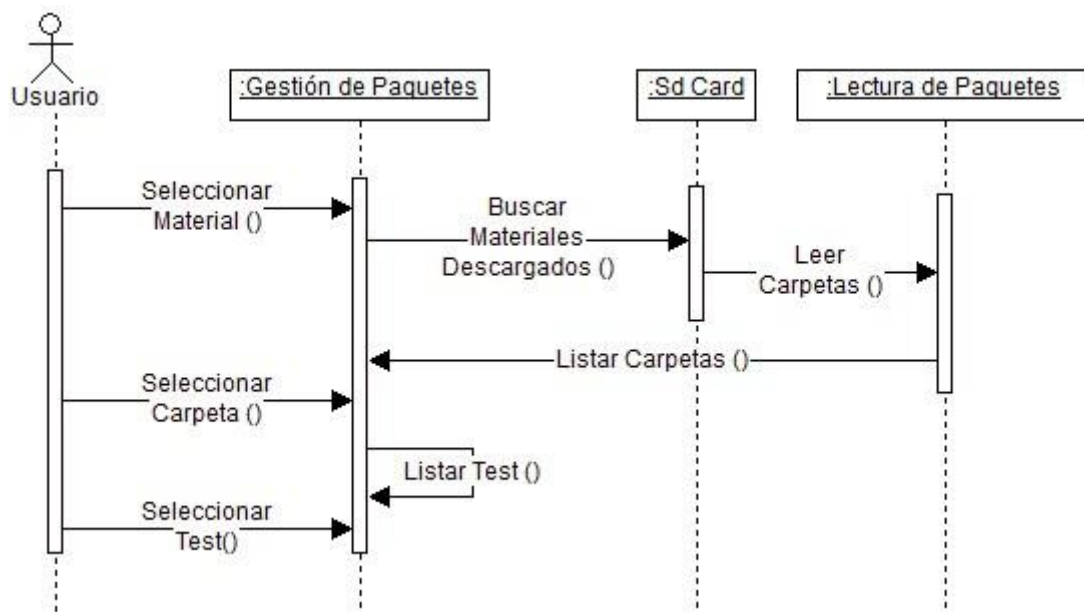


Figura 18. Diagrama de Secuencia Seleccionar Material

- Navegar por el test: Permite al usuario, pulsando “→” o “←”, pasar de ejercicio en ejercicio dentro de un mismo test.

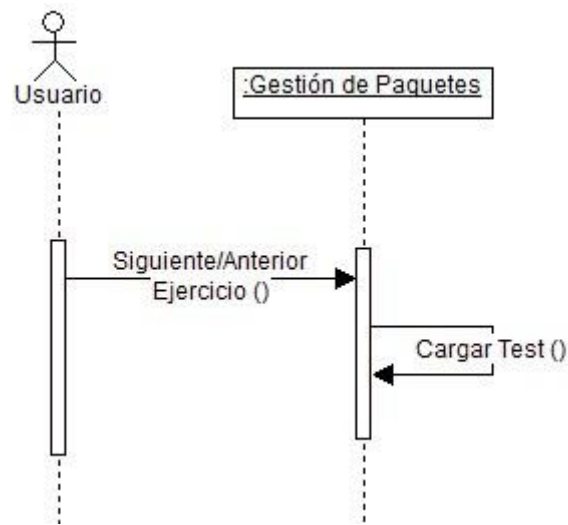


Figura 19. Diagrama de Secuencia Navegar por el Test

- Ver Anotaciones: Permite al usuario, pulsando la pantalla, ver el contenido completo de la anotación.

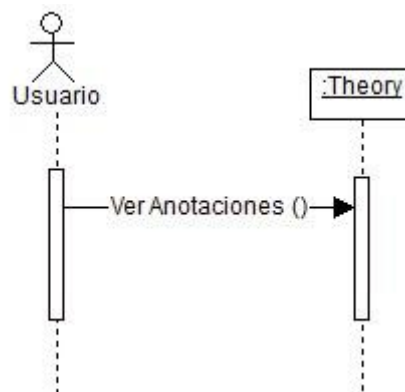


Figura 20. Diagrama de Secuencia Ver Anotaciones

- Consultar Puntuación: Permite al usuario, pulsando “show score”, ver la calificación obtenida de los ejercicios contestados del test.

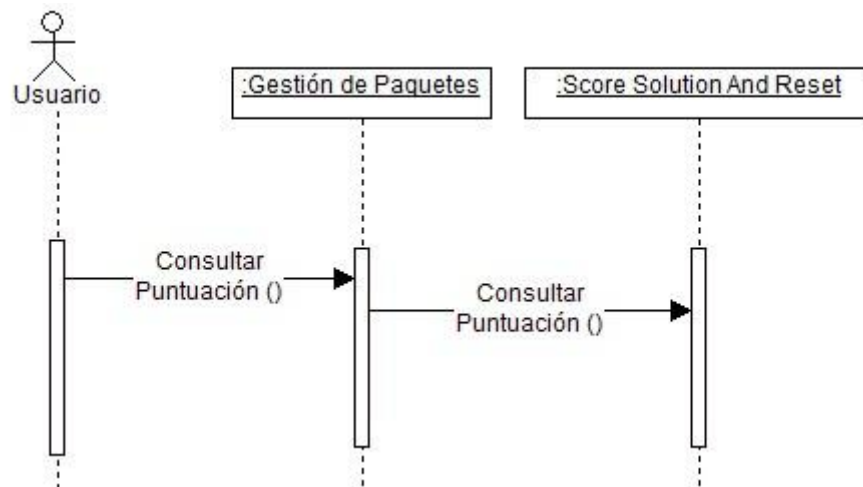


Figura 21. Diagrama de Secuencia Consultar Puntuación

- Consultar Soluciones: Permite al usuario, pulsando “show solution”, ver la solución correcta de los ejercicios contestados del test.

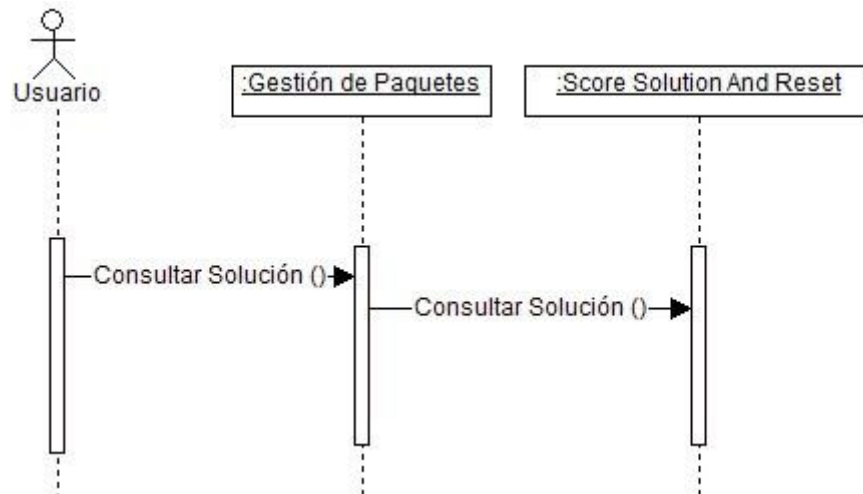


Figura 22. Diagrama de Secuencia Consultar Soluciones

- Corregir Test: Permite al usuario, pulsando “check”, saber si la respuesta elegida es correcta o errónea. En este caso, dependiendo del ejercicio en el que se encuentre, estará involucrado un componente u otro del módulo de despliegue.

- Si el usuario se encuentra en un ejercicio tipo “audio choice”

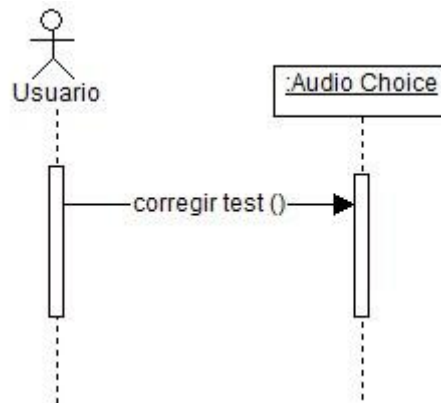


Figura 23. Diagrama de Secuencia Corregir Test Audio Choice

- Si el usuario se encuentra en un ejercicio tipo “drag and drop”

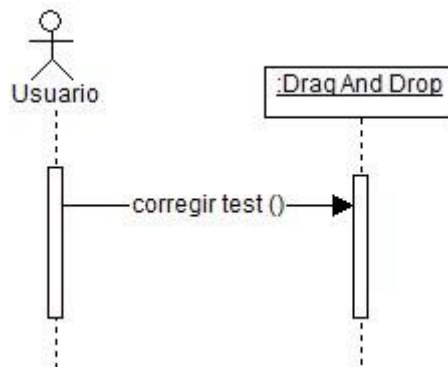


Figura 24. Diagrama de Secuencia Corregir Test Drag And Drop

- Si el usuario se encuentra en un ejercicio tipo “multiple fill the gap” o “fill the gap” o “multiple choice”.

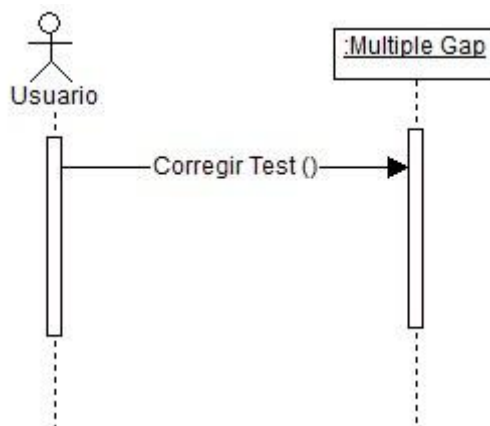


Figura 25. Diagrama de Secuencia Corregir Test Multiple Gap

- Si el usuario se encuentra en un ejercicio tipo “Word order”

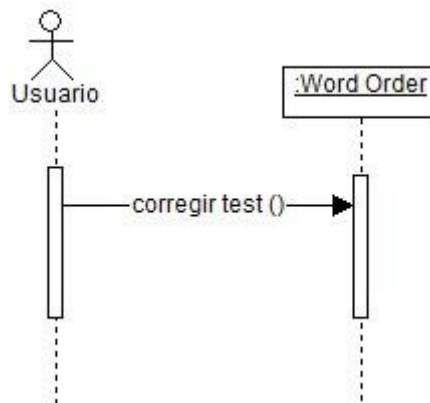


Figura 26. Diagrama de Secuencia Corregir Test Word Order

- Escuchar Audio: Permite al usuario, pulsando “play audio”, escuchar el audio asociado al ejercicio. En este caso, dependiendo del ejercicio en el que se encuentre, estará involucrado un componente u otro del módulo de despliegue.

- Si el usuario se encuentra en un ejercicio tipo “audio”

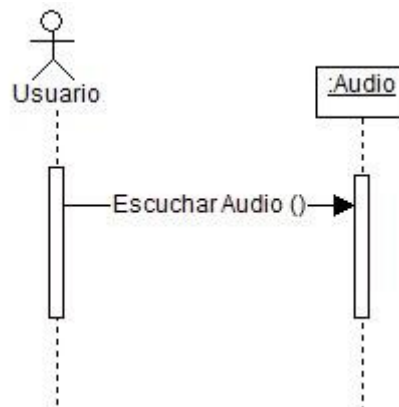


Figura 27. Diagrama de Secuencia Escuchar Audio

- Si el usuario se encuentra en un ejercicio tipo “audio choice”

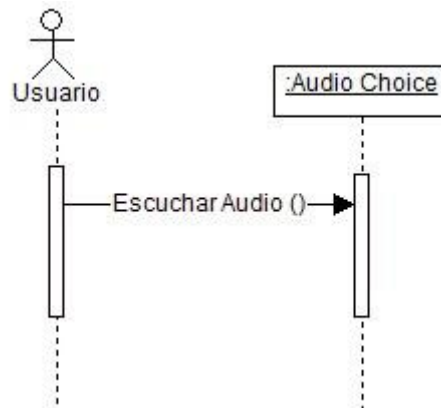


Figura 28. Diagrama de Secuencia Escuchar Audio

- Reiniciar Test: Permite al usuario, pulsando “reset”, volver a realizar el test borrando las respuestas, resultados y soluciones de los ejercicios contestados.

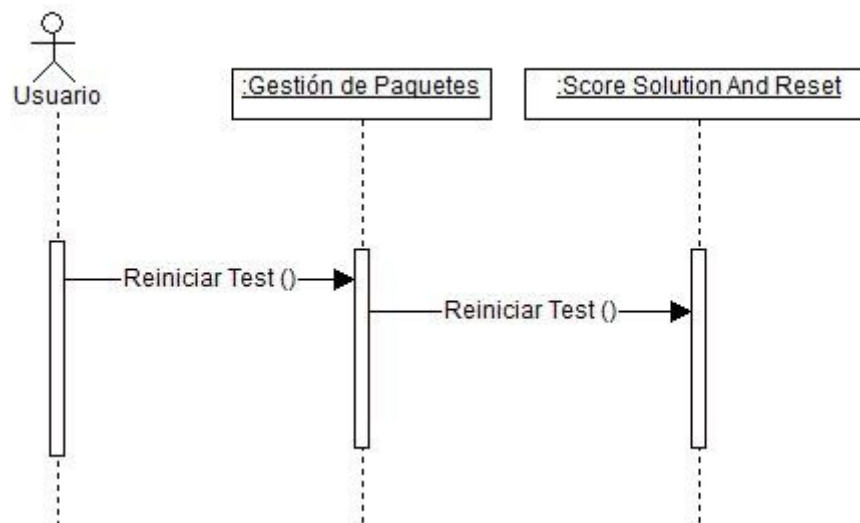


Figura 29. Diagrama de Secuencia Reiniciar Test

- Salir Menú Principal: Permite al usuario, pulsando “exit” volver a la pantalla principal de la aplicación.

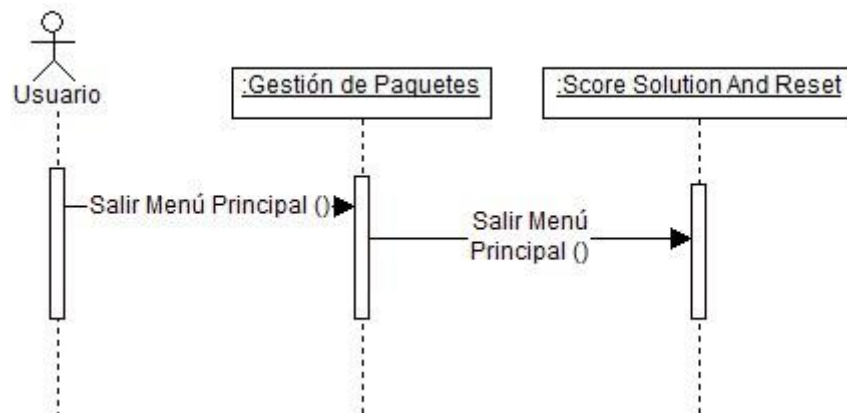


Figura 30. Diagrama de Secuencia Salir Menú Principal

- Salir: Permite al usuario, pulsando “salir”, cerrar la aplicación cuando desee.

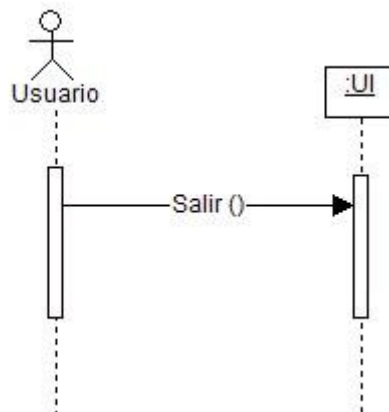


Figura 31. Diagrama de Secuencia Salir

5.3 Prototipos de la Interfaz de Usuario

En la fase de diseño, el propósito del prototipo es obtener una primera versión de la apariencia de la interfaz de usuario así como de la funcionalidad incluida (mostrar las ventanas, su navegación, interacción, controles y botones). Con esto se pretende que el cliente tenga una primera toma de contacto con la futura aplicación antes de su desarrollo final, para así reducir o eliminar todas aquellas disconformidades y cambios en fases futuras.

La herramienta utilizada para ello es **Pencil** v1.2, proyecto de código libre que tiene como objetivo desarrollar una herramienta para crear prototipos de interfaces de usuario.

A continuación se muestran los prototipos de bajo nivel realizados para las distintas pantallas de la aplicación.

5.3.1 Pantalla Menú Principal

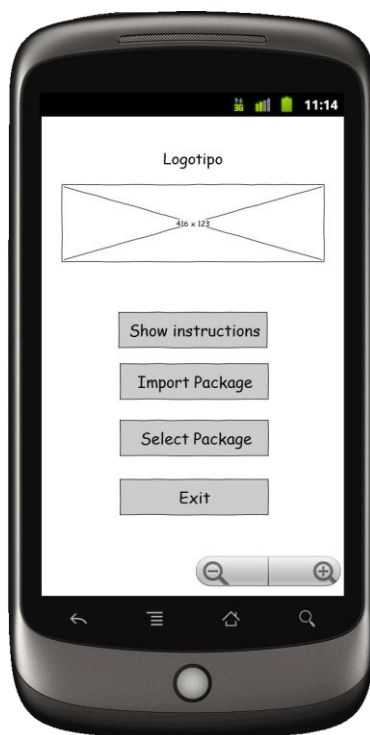


Figura 32. Prototipo Pantalla Menú Principal

El usuario, al ejecutar la aplicación accede a una pantalla cuyo contenido es el menú principal de la herramienta. Este menú aparece únicamente en la pantalla principal de la aplicación. Mediante los botones que se muestran el usuario puede acceder a las funcionalidades básicas como la importación de material educativo, la

visualización de las instrucciones de manejo así como la selección de un test para su realización y salir de la aplicación. Los botones contienen un nombre descriptivo de la funcionalidad que proporcionan.

5.3.2 Pantalla Descarga Material Educativo

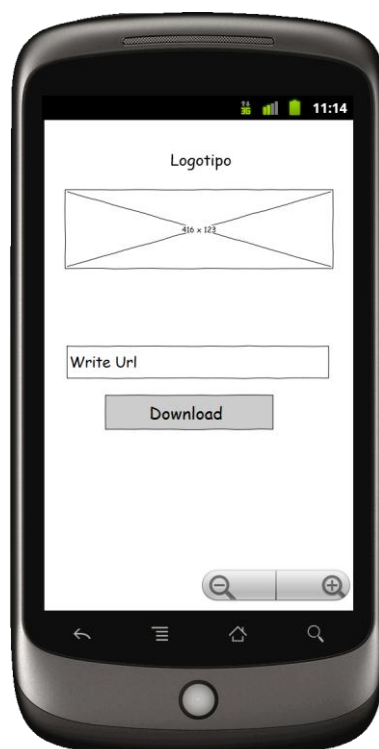


Figura 33. Prototipo Pantalla Descarga Material Educativo

Tras pulsar el botón de importar nuevo material educativo, se accede a la pantalla mostrada. En ella, el usuario deberá introducir la url donde esté ubicado el material para su descarga y posteriormente pulsar el botón para comenzar la descarga. La aplicación se encargará de descargar, descomprimir y guardar dicho material en la carpeta correspondiente. El botón contiene un nombre descriptivo de la funcionalidad que proporciona. Los listados contienen un nombre descriptivo, primero de los materiales existentes, y después del contenido del material seleccionado.

5.3.3 Pantalla Listar Material Educativo

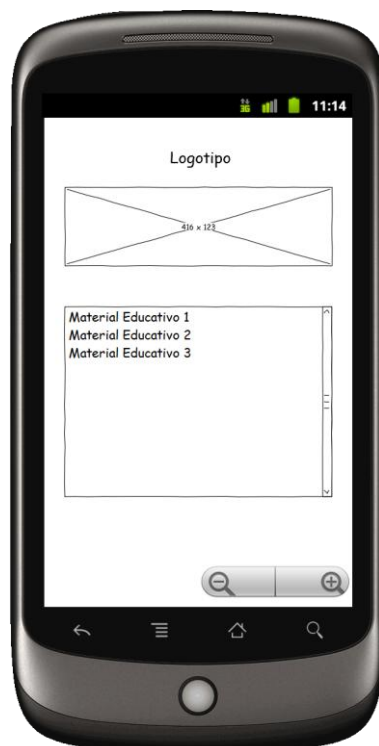


Figura 34. Prototipo Pantalla Listar Material Educativo

El usuario tras pulsar el botón de seleccionar un material educativo accederá a la pantalla mostrada donde se lista el contenido de la carpeta del programa donde se ubican los materiales descargados. En ella, es posible seleccionar un material, dando lugar a una pantalla idéntica a la anterior, pero esta vez con el contenido del material seleccionado. El usuario no tendrá más que seleccionar el test que desea realizar para empezar.

5.3.4 Pantallas de Ejercicios



Figura 35. Prototipo Pantallas de Ejercicios

Tras seleccionar el test que el usuario desea realizar, comienza la ejecución del mismo con el primer ejercicio que contenga. Se mostrara al usuario la parte denominada común en la figura anterior que contiene en la parte superior de la pantalla el tipo de ejercicio a realizar, y en la parte inferior se muestra un menú con un máximo de 5 botones. Dichos botones sirven para navegar dentro del test (es decir, pasar de ejercicio en ejercicio), para navegar dentro de un ejercicio (es decir, pasar de pregunta dentro de un mismo ejercicio) y para corregir la respuesta seleccionada del ejercicio.

Según el tipo de ejercicio en el que se encuentre el usuario, la funcionalidad de la parte intermedia de la pantalla irá cambiando.

Para los ejercicios que contienen preguntas de rellenar el hueco (mostrado en la figura como ejercicios con gap), la parte intermedia de la pantalla mostrará el enunciado de la pregunta y dará la posibilidad al usuario de seleccionar una o varias respuestas.

Para los ejercicios que contengan un archivo de audio (mostrados en la figura como ejercicios con audio) la parte intermedia mostrará un botón para reproducir el archivo y un control de volumen para que el usuario lo ajuste a su gusto. Además, si el ejercicio contiene preguntas, mostrará el enunciado y las preguntas que contengan de igual forma que los ejercicios con gap.

Para los ejercicios de ordenar la frase (mostrados en la figura como ejercicios de ordenar), la parte intermedia muestra la frase desordenada y un cuadro de texto donde el usuario debe introducir la respuesta.

Por último, para los ejercicios con contenido introductorio y de teoría, la parte intermedia de la pantalla podrá mostrar tantos cuadros de texto e imágenes como se

necesiten. Además, el texto podrá contener anotaciones ocultas que se muestran como un texto con hipervínculo, dando lugar a la aparición de una ventana emergente con el contenido de la anotación.

Los botones de esta pantalla contienen un nombre o un icono descriptivo de la funcionalidad que proporcionan.

5.3.5 Pantalla de Soluciones y Puntuación

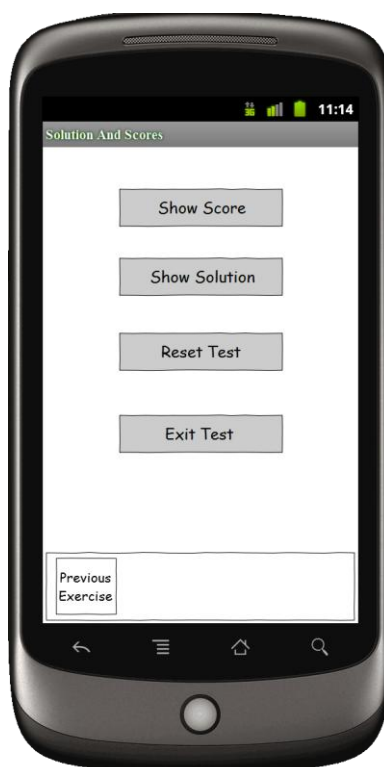


Figura 36. Prototipo Pantalla de Soluciones y Puntuación

Tras terminar los ejercicios que componen el test, el usuario accede a la pantalla mostrada donde se muestra un menú. Este menú aparece únicamente tras la finalización de un test. Mediante los botones que se muestran el usuario puede acceder a las funcionalidades básicas como mostrar los resultados del test realizado, mirar las soluciones, resetear el test para hacer de nuevo y volver al menú principal.

Capítulo 6

IMPLEMENTACIÓN

El propósito de este capítulo es describir los aspectos más relevantes con respecto a la implementación de la aplicación `MobileTestAndroidPlayer`. Es decir, no sólo se mostrará el código más importante, sino que también sirve como una explicación sobre cómo funcionan las aplicaciones para Android. Destacar que las funcionalidades básicas como el paso de parámetros entre actividades u otros temas triviales no serán explicadas, puesto que para ello existen multitud de tutoriales en la página de desarrollo de Android “Android Developers”.

Así mismo, los aspectos más relevantes con respecto a la implementación mencionados esperan poder ayudar a otros desarrolladores a ampliar las capacidades de `MobileTestAndroidPlayer`.

6.1 Descarga Material Educativo

Una de las funcionalidades básicas de la aplicación no es otra que la descarga de un archivo comprimido en formato ZIP desde una dirección web y lo descomprime.

La descarga del paquete desde una URL ha sido realizada mediante la función `DownloadFromUrl`, que activa la conexión a internet y descarga el paquete en la tarjeta de memoria dándole el nombre que contenga en la dirección web.

```
public void DownloadFromUrl(String urlfile) {
    int count;
    try {
        URL url = new URL(urlfile);
        URLConnection conexion = url.openConnection();
        conexion.connect();
        InputStream input = new BufferedInputStream(url.openStream());
        OutputStream output = new FileOutputStream(
            Environment.getExternalStorageDirectory().getAbsolutePath()+"/"+Nombre);
        byte data[] = new byte[1024];
        while ((count = input.read(data)) != -1) {
            output.write(data, 0, count);
        }
        output.flush();
        output.close();
        input.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Código 05. Código Función DownloadFromUrl

La descompresión del archivo descargado la realiza la función `unzip`, que se encarga de recorrer el archivo comprimido viendo si lo que está leyendo es un directorio o un archivo. En el caso de que esté leyendo un directorio, crea la carpeta para posteriormente meter los archivos que contenga. Todos los archivos y directorios leídos los meterá dentro de la carpeta del programa en la tarjeta de memoria del dispositivo.

```

public void unzip() {
    try {
        FileInputStream fin = new FileInputStream(_zipFile);
        ZipInputStream zin = new ZipInputStream(fin);
        ZipEntry ze = null;
        while ((ze = zin.getNextEntry()) != null) {
            Log.v("Decompress", "Unzipping " + ze.getName());
            if(ze.isDirectory()) {
                _dirChecker(ze.getName());
            } else {
                FileOutputStream fout = new FileOutputStream(_location + ze.getName());
                for (int c = zin.read(); c != -1; c = zin.read()) {
                    fout.write(c);
                }
                zin.closeEntry();
                fout.close();
            }
        }
        zin.close();
    } catch(Exception e) {
        Log.e("Decompress", "unzip", e);
    }
}

```

Código 06. Código Función Unzip

El principal problema que surgió a la hora de descargar y seguidamente descomprimir, era que no se respetaba un orden secuencial de ejecución. Es decir, que la descarga y la descompresión se hacían simultáneamente, no llegando a realizar bien ninguna de las dos funcionalidades.

La solución a la que se llegó era incluir la llamada a la función dentro de un hilo o thread, para llevar un orden de ejecución secuencial.

Primero llama a la función que descarga el archivo comprimido, definimos la ubicación del archivo comprimido y donde debe guardarse descomprimido y por último nos vamos a la actividad donde está el método para descomprimir.

```

Thread checkUpdate = new Thread() {
    public void run() {
        DownloadFromUrl(Uri);
        String zipFile = Environment.getExternalStorageDirectory().getAbsolutePath()
            + "/" + Nombre;
        String unzipLocation = Environment.getExternalStorageDirectory().getAbsolutePath()
            + "/MobileTestAndroidPlayer/"+nomb+"/";
        handler.sendMessage(0);
        cambiarVentana (zipFile , unzipLocation);
    }
};
checkUpdate.start();

```

Código 07. Código del Thread para DescargarMaterial Educativo

En ella, descomprimos el paquete y volvemos al menú principal de la aplicación.

```
Thread checkUpdate = new Thread() {
    public void run() {
        Decompress d = new Decompress(zipFile, unzipLocation);
        d.unzip();
        handler.sendMessage(0);
        Intent intent = new Intent(Descomprimir.this, Principal.class);
        intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        startActivity(intent);
    }
};
checkUpdate.start();
```

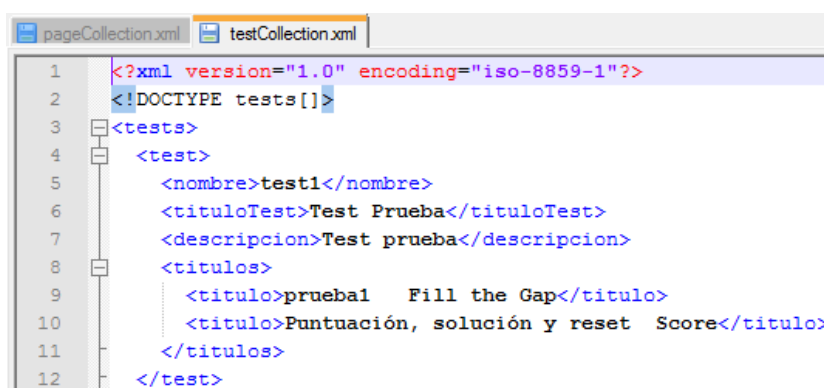
Código 08. Código del Thread para Descomprimir Material Educativo

6.2 Lectura de los Archivos XML

Uno de los aspectos más importantes de la aplicación es la lectura de los archivos “testcollection.xml”, que contiene un listado de todos los test con los ejercicios de los que se componen, y el archivo “pagecollection.xml” que contiene un listado con el contenido de cada ejercicio. Estos y más archivos los podemos encontrar en cada uno de los materiales educativos descargados.

Sin la lectura de estos dos archivos, la aplicación no podría realizar ninguna de las funcionalidades de las que consta.

Como se vio en el capítulo 2 sección 4 “XML”, XML es un metalenguaje cuya labor es definir unas reglas que deberá cumplir un lenguaje de marcas. Las marcas o etiquetas definen el principio y el fin de un contenido. En el archivo “testcollection” se definen etiquetas que delimitan el contenido de cada test y dentro de cada uno, una serie de etiquetas donde podemos encontrar el nombre y título del test, una descripción y los ejercicios de los que se componen llamados títulos. La siguiente figura muestra dicho archivo con las etiquetas que definen un test.



Código 09. Código XML del Archivo Testcollection

Cuando el usuario desee realizar un test, seleccionará un material educativo y la aplicación leerá su archivo “testcollection” para mostrar los test que contiene. Pero, ¿Cómo lee la aplicación este archivo?

Para ello, en Android existen varias formas de leer y escribir datos en formato XML. El elegido para la aplicación es el modelo XmlPull. En él, el tratamiento de un XML se basa en un parser o analizador que a medida que lee secuencialmente el documento va generando eventos con la información de cada elemento leído. Los eventos que se producen son los siguientes:

- Start_Document (): comienza el documento XML.
- End_Document (): termina el documento XML.
- Start_Tag (): comienza una etiqueta XML.
- End_Tag (): termina una etiqueta XML.

Nuestro trabajo consistirá por tanto en implementar las acciones necesarias a ejecutar para cada uno de los eventos posibles que se pueden generar durante la lectura del documento XML. Esas acciones consistirán en guardar en una lista de tests, todo el contenido de cada uno de los test que se incluyan.

```
public List<test> parse()
{
    List<test> tests = null;
    int n=0;
    XmlPullParser parser = Xml.newPullParser();
    try
    {
        parser.setInput(this.getInputStream(), "ISO-8859-1");
        int evento = parser.getEventType();
        test testActual = null;
        while (evento != XmlPullParser.END_DOCUMENT)
        {
            String etiqueta = null;
            switch (evento)
            {
                case XmlPullParser.START_DOCUMENT:
                    tests = new ArrayList<test>();
                    break;
                case XmlPullParser.START_TAG:
                    etiqueta = parser.getName();
                    if (etiqueta.equals("test"))
                    {
                        testActual = new test();
                    }
                    else if (testActual != null)
                    {
                        if (etiqueta.equals("nombre"))
                        {
                            testActual.setNombre(parser.nextText());
                        }
                        else if (etiqueta.equals("tituloTest"))
                        {
                            testActual.setTituloTest(parser.nextText());
                        }
                        else if (etiqueta.equals("descripcion"))
                        {
                            testActual.setDescripcion(parser.nextText());
                        }
                        else if (etiqueta.equals("titulos"))
                        {
                            n=0;
                        }
                        else if (etiqueta.equals("titulo"))
                        {
                            testActual.getTitulos().put(n, parser.nextText());
                            n++;
                        }
                    }
                    break;
            }
        }
    }
}
```

Código 10. Código del Parser del Archivo Testcollection

```

        case XmlPullParser.END_TAG:
            etiqueta = parser.getName();
            if (etiqueta.equals("test") && testActual != null)
            {
                tests.add(testActual);
                break;
            }
            evento = parser.next();
        }
    }
}

```

Código 11. Código del Parser del Archivo Testcollection

Una vez llegado a este paso, la aplicación pasará a leer el archivo “pagecollection” que contiene la información de cada uno de los ejercicios de los test. Al igual que para el archivo anterior, se crea un parser similar, donde guardaremos una lista con todos los ejercicios. Cada ejercicio tiene una etiqueta contenido, y dependiendo del tipo de ejercicio que sea, guardará dentro de ella una información u otra.

```

else if (etiqueta.equals("contenido"))
{
    contenido = new contenido();
    x=0;y=0;z=0;k=0;t=0;q=0;a=0;b=0;
}
else if (etiqueta.equals("pregunta"))
{
    pregunta = new pregunta();
}
else if (etiqueta.equals("enunciado"))
{
    pregunta.setEnunciado_pregunta(parser.nextText());
}
else if (etiqueta.equals("respuestas"))
{
    n=0;
}
else if (etiqueta.equals("respuesta"))
{
    pregunta.getRespuestas_pregunta().put(n, parser.nextText());
    n++;
}
else if (etiqueta.equals("correcta"))
{
    pregunta.setCorrecta_pregunta(parser.nextText());
}
else if (etiqueta.equals("texto"))
{
    contenido.setTexto(parser.nextText());
}
else if (etiqueta.equals("titulo_teoría"))
{
    contenido.getTitulo_teoría().put(b, parser.nextText());
    b++;
    contenido.getTheory().put(t, "titulo teoría");
    t++;
}
else if (etiqueta.equals("uso"))
{
    contenido.getUsos().put(a, parser.nextText());
    a++;
    contenido.getTheory().put(t, "uso");
    t++;
}
else if (etiqueta.equals("regla"))
{
    contenido.getReglas().put(q, parser.nextText());
    q++;
    contenido.getTheory().put(t, "regla");
    t++;
}
else if (etiqueta.equals("imagen"))
{
    contenido.setImagen(parser.nextText());
    contenido.getTheory().put(t, "imagen");
    t++;
}
else if (etiqueta.equals("audio"))
{
    if (paginaActual.getTipo() == "Audio") {
        contenido.setAudio(parser.nextText());
    } else {
        contenido.setAudio_choice(parser.nextText());
    }
}
}
}

```

Código 12. Código del Parser del Archivo Pagecollection

6.3 Listados de Contenidos

El usuario no podrá seleccionar ningún ejercicio del material educativo que desee si no dispone de una herramienta que le permita visualizarlos y seleccionarlos. Para ello, se decidió usar un ArrayAdapter con control de selección ListView.

Nuestro ArrayAdapter provee de datos al control de selección ListView a partir de un array de objetos de cualquier tipo, que en nuestro caso es un array con los nombres de los materiales educativos. El ListView por su parte, muestra al usuario la lista de materiales seleccionables directamente sobre el propio control, sin crear una lista emergente.

```
PackageAdapter adapter = new PackageAdapter(this,
    R.layout.list_item, Paquetes);

ListView lstOpciones = (ListView)findViewById(R.id.listView);
lstOpciones.setAdapter(adapter);
```

Código 13. Código ArrayAdapter

La primera línea de código lo que hace es crear el adaptador en el activity en el que nos encontramos, mostrando el contenido del array Paquetes en forma de lista. El tipo de datos PackageAdapter es el encargado de crearlo. Se trata de una nueva activity donde se genera el adaptador por medio de la función getView cada vez que hay que dibujar la lista.

De esta forma, si en un futuro se desea que la lista de elementos contenga más datos a parte del nombre de los paquetes como puede ser una imagen o cualquier otro elemento, sólo habría que añadirle dicha funcionalidad sin tocar nada del propio código.

Igualmente, el código es análogo al comentado cuando se necesita mostrar el contenido de un material educativo. En este caso, en vez de mostrar el nombre de los materiales, mostrará los nombres de los test que contiene el material seleccionado por el usuario.

Por último destacar la posibilidad de borrar un material educativo. Para que el usuario no acumule materiales ya realizados y que no desea conservar, el usuario, realizado una pulsación larga sobre el material que desee podrá borrar dicho paquete de la tarjeta de memoria del terminal móvil.

```

lstOpciones.setOnItemLongClickListener(new AdapterView.OnItemLongClickListener() {
    @Override
    public boolean onItemLongClick(AdapterView<?> parent, android.view.View v,
        int position, long id) {
        final int Borrar = position;
        alertbox.setMessage("Delete Package?");
        alertbox.setCancelable(false);
        alertbox.setPositiveButton("Yes", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int id) {
                File file = new File(Environment.getExternalStorageDirectory().getAbsolutePath()
                    + "/MobileTestAndroidPlayer/"+Paquetes[Borrar]);
                deleteDirectory(file);
                Intent intent = new Intent(Iniciar.this, Iniciar.class);
                intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                startActivity(intent);
            }
        });
    }
});

```

Código 14. Código Borrar Material Educativo

6.4 Funcionalidades en los Ejercicios

Las funcionalidades de los ejercicios se consiguen gracias al manejo de las propiedades de elementos simples como radiobutton, boolean, string o arrays, elementos no destacables por su gran amplitud de tutoriales en la red.

Si merece mención, en los ejercicios que contienen un archivo de audio, su reproducción y el control de volumen del mismo. Para reproducir el archivo de audio se necesita del elemento MediaPlayer para controlar la reproducción de archivos de audio/video y streams. En Path_To_File introducimos la ruta donde se encuentra el archivo.

```

Button boton = (Button) findViewById(R.id.b_repro);
boton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(start==true){
            snd.stop();
        }
        snd = new MediaPlayer();
        try{
            Globales appState = ((Globales)getApplicationContext());
            String paquete = appState.getNombrePaquete();
            String PATH_TO_FILE = Environment.getExternalStorageDirectory().getAbsolutePath()
                +"/MobileTestAndroidPlayer/"+paquete+"/Audio/"+audio;
            snd.setDataSource(PATH_TO_FILE);
            snd.prepare();
        }catch(IOException ie){ie.printStackTrace();}
        snd.start();
        start=true;
    }
});

```

Código 15. Código Botón Reproducir Audio

La clase AudioManager nos proporciona el acceso al control de volumen. Con ella y un seekBar como elemento visual el usuario controla el volumen según sus necesidades.

```

audioManager = (AudioManager) getSystemService (Context.AUDIO_SERVICE);
int maxVolume = audioManager.getStreamMaxVolume (AudioManager.STREAM_MUSIC);
int curVolume = audioManager.getStreamVolume (AudioManager.STREAM_MUSIC);
SeekBar volControl = (SeekBar) findViewById(R.id.seekBar);
volControl.setMax(maxVolume);
volControl.setProgress(curVolume);
volControl.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
    @Override
    public void onProgressChanged(SeekBar arg0, int arg1, boolean arg2) {
        // TODO Auto-generated method stub
        audioManager.setStreamVolume (AudioManager.STREAM_MUSIC, arg1, 0);
    }
});

```

Código 16. Código Control de Volumen

Por último, destacar la clase Globales donde se declaran todas aquellas variables globales que necesitamos durante el transcurso de un test. En ella tenemos todos los métodos para guardar dichas variables.

Algunas de las variables que se guardan son las preguntas contestadas, el total de preguntas, la puntuación del ejercicio, todas ellas necesarias cuando el usuario termine el test y quiera consultarlas. O en material en el que nos encontramos para cuando necesitemos reproducir un audio saber donde se encuentra.

```

public class Globales extends Application {

    private List<test> I_test = new ArrayList<test>();
    private List<pagina> I_pagina = null;
    private int QuieroElTest = -1;
    private String NombrePaquete = "";
    private int tam =0;
    private int Score[] = null;
    private String Titulos[] = null;
    private int Total[] = null;
    private String Solution[] = null;
    private Hashtable Contestadas = new Hashtable();

    private int N_Drag = -1;

```

Código 17. Código del Archivo Globales

6.5 Pruebas

El objetivo de estas pruebas es verificar que los subsistemas funcionan correctamente, que son capaces de interactuar unos con otros y que todo funciona correctamente.

Para una mejor descripción de las pruebas se utiliza la tabla mostrada a continuación:

PRUEBA XX	
Objetivo	Funcionalidad a probar
Precondiciones	condiciones para realizar la prueba
Postcondiciones	Resultados de la prueba
Descripción	Pasos a seguir

Tabla 72. Tabla de Pruebas

6.5.1 Descarga Material Educativo

PRUEBA 01	
Objetivo	Descargar Material Educativo
Precondiciones	El usuario debe encontrarse en la pantalla “import new package” e introducir una url.
Postcondiciones	Se descarga y descomprime el material en la carpeta de la aplicación.
Descripción	El usuario debe encontrarse en la pantalla “import new package” e introducir una url. Si la url no es correcta, o no alberga un archivo comprimido ZIP mostrará un mensaje de error. Si todo es correcto muestra los mensajes de descarga, descompresión y finalización.

Tabla 73. Prueba Descargar Material Educativo

Con esta prueba conseguimos demostrar el correcto funcionamiento de dicha funcionalidad. La aplicación consigue descargar el paquete, descomprimirlo y guardarlo en una carpeta dentro de la propia carpeta de la aplicación.

6.5.2 Lectura de los Archivos XML

PRUEBA 02	
Objetivo	Lectura archivos XML
Precondiciones	Tener descargado al menos un material educativo y encontrarse en la pantalla “select package and star”
Postcondiciones	La aplicación lee los archivos y los almacena para un uso posterior.
Descripción	EL usuario debe tener descargado al menos un material educativo y encontrarse en la pantalla “select package and star”. El usuario selecciona un material educativo y la aplicación deberá leer el archivo Testcollection.XML, mostrando los tests disponibles. El usuario seleccionará un test. Posteriormente la aplicación deberá leer el archivo pagecollection.xml para ir consultándolo cuando se necesite.

Tabla 74. Prueba Lectura Archivos XML

Con esta prueba se consigue leer dos de los archivos que contiene el material educativo. Estos archivos contienen la colección de test disponibles (testcollection.xml) y el contenido de todos los test (pagecollection.xml). Sin la lectura de estos no se podría implementar las funcionalidades de los ejercicios.

Cabe destacar que cuando probamos su correcto funcionamiento, no disponíamos de un menú para seleccionar el material y el test a realizar, por lo que se probó de forma interna. Es decir, dentro del entorno de desarrollo utilizando su modo debug para comprobar línea a línea que leía todo el archivo de forma correcta.

6.5.3 Listados de Contenidos

PRUEBA 03	
Objetivo	Listado de Contenidos
Precondiciones	Tener descargado al menos un material educativo y encontrarse en la pantalla “select package and star”
Postcondiciones	Se muestra el contenido de la carpeta de la aplicación
Descripción	El usuario debe tener descargado al menos un material educativo y encontrarse en la pantalla “select package and star”. Primero la aplicación debe mostrar el contenido de su carpeta con los materiales disponibles. Al seleccionar uno, debe mostrar los nombres de los test que contiene el archivo testcollection de la carpeta seleccionada.

Tabla 75. Prueba Listado de contenidos

Sin que se dé cuenta el usuario conseguimos navegar por el contenido de la carpeta de la aplicación, ubicada en la tarjeta de memoria del dispositivo móvil.

6.5.4 Funcionalidades de los Ejercicios

PRUEBA 04	
Objetivo	Corregir Ejercicio
Precondiciones	Tener descargado al menos un material, haber seleccionado un test de un material y seleccionar una respuesta del ejercicio.
Postcondiciones	Muestra si la respuesta es correcta o errónea
Descripción	El usuario debe tener descargado al menos un material, haber seleccionado un test de un material y seleccionar una respuesta del ejercicio. Pulsa el botón “check” y la aplicación deberá comparar la respuesta con la solución. Si es correcta debe colorear el texto de verde y mostrar una imagen positiva. Si es errónea debe colorear el texto de rojo y mostrar una imagen negativa. Además deberá bloquear el ejercicio para que no se pueda realizar de nuevo.

Tabla 76. Prueba Corregir Ejercicio

Con esto conseguimos probar además de que el usuario pueda ver de forma visual su acierto o error, que la aplicación recupera los datos leídos del archivo pagecollection y que lee bien su contenido.

PRUEBA 05	
Objetivo	Texto con estilos html
Precondiciones	Tener descargado al menos un material, y haber seleccionado un test de un material.
Postcondiciones	Se muestra el texto con los estilos propios de html si estos contienen dichas marcas.
Descripción	El usuario debe tener descargado al menos un material, y haber seleccionado un test de un material. Si el texto mostrado contiene etiquetas de estilo, la aplicación debe mostrar el texto con el formato pedido.

Tabla 77. Prueba Texto con estilos html

PRUEBA 06	
Objetivo	Siguiente/Anterior Ejercicio o Gap
Precondiciones	Tener descargado al menos un material, haber seleccionado un test de un material.
Postcondiciones	Se pasará al ejercicio/gap siguiente/anterior
Descripción	El usuario debe tener descargado al menos un material, haber seleccionado un test de un material. El usuario pulsará una de las teclas de movimiento y la aplicación debe cambiar a la pantalla indicada.

Tabla 78. Prueba Siguiente/Anterior Ejercicio o Gap

Con esta prueba conseguimos que el usuario navegue con perfecta fluidez, por todas y cada una de las pantallas del test en el que se encuentra. Para así comprobar que salta a la pantalla adecuada.

PRUEBA 07	
Objetivo	Reproducir audio
Precondiciones	Tener descargado al menos un material, haber seleccionado un test de un material y encontrarse en un ejercicio de audio o audio choice.
Postcondiciones	Reproduce el audio que contiene.
Descripción	El usuario debe tener descargado al menos un material, haber seleccionado un test de un material y encontrarse en un ejercicio de audio o audio choice. Pulsa el botón “play audio” y la aplicación debe ser capaz de buscar en la carpeta del material educativo del test el archivo de audio para reproducirlo.

Tabla 79. Prueba Reproducir audio

Junto a esta prueba se realiza también la prueba del control de volumen, puesto que si no se reproduce el audio no podemos ver si el control funciona. Para ello si el usuario pulsa sobre la barra de volumen, la aplicación debe ser capaz de aumentar/disminuir el volumen.

PRUEBA 08	
Objetivo	Expandir Teoría
Precondiciones	Tener descargado al menos un material, haber seleccionado un test de un material y encontrarse en un ejercicio de tipo “theory”.
Postcondiciones	Se muestra el contenido oculto.
Descripción	Si el usuario pulsa sobre un elemento de teoría que contenga un “+”, la aplicación debe expandir el contenido oculto para mostrarlo.

Tabla 80. Prueba Expandir Teoría

Junto con la posible expansión de los elemento de teoría con contenido oculto, estos elementos también pueden incluir anotaciones ocultas. La prueba para su correcto funcionamiento es idéntica a la anterior.

Al realizar la prueba se comprueba que la expansión del contenido no funciona de manera óptima y se procede a su corrección.

PRUEBA 08	
Objetivo	Introducir texto
Precondiciones	Tener descargado al menos un material, haber seleccionado un test de un material y encontrarse en un ejercicio de tipo “Word order”.
Postcondiciones	Se introduce el texto en la caja de texto.
Descripción	El usuario debe tener descargado al menos un material, haber seleccionado un test de un material y encontrarse en un ejercicio de tipo “Word order”. Pula sobre la caja de texto y la aplicación debe mostrar el teclado para la escritura y recoger el texto en la caja.

Tabla 81. Prueba Introducir texto

Al realizar la prueba y dejar el texto en blanco, al comprobar salta un mensaje de error avisando que se debe escribir una respuesta pero posteriormente cuando se desea escribirla está bloquea la escritura. Se debe a un error en el código que se procede a modificar.

PRUEBA 09	
Objetivo	Mostrar soluciones
Precondiciones	Tener descargado al menos un material, haber seleccionado un test de un material y encontrarse en el menú “score solution and reset”
Postcondiciones	Se muestra las soluciones de las preguntas respondidas
Descripción	El usuario debe tener descargado al menos un material, haber seleccionado un test de un material y encontrarse en el menú “score solution and reset”. Al pulsar sobre “show solution” la aplicación debe mostrar las soluciones de los ejercicios. Se debe comprobar que las soluciones son las esperadas.

Tabla 82. Prueba Mostrar soluciones/puntuaciones

De igual forma, se realiza una prueba para mostrar las puntuaciones que sigue el mismo procedimiento que la anterior. Se deberá comprobar que las puntuaciones son correctas.

Capítulo 7

MANUAL DE USUARIO Y DE INSTALACIÓN

7.1 Manual de Usuario

El objetivo del presente apartado es mostrar las diferentes interfaces diseñadas para la aplicación, así como comentar su funcionalidad y cómo el usuario puede interactuar con ellas. De esta forma, obtendremos un pequeño manual de usuario.

Al iniciar la aplicación, se muestra una interfaz que contiene en la cabecera el logotipo y debajo de éste el menú principal, permitiendo al usuario seleccionar la funcionalidad a la que desea acceder.



Figura 37. Pantalla Principal MobileTestAndroidPlayer

Como puede observarse en la interfaz del menú principal, el usuario tiene cuatro opciones de entre las que puede elegir. Es recomendable la primera vez que el usuario inicie la aplicación, se dirija a la opción “select package and star” para que se genere la carpeta de destino de los materiales educativos que se descarguen durante su uso. Y seguidamente vaya a la opción “import new package” para descargar un material educativo para poder disfrutar de toda la funcionalidad de la aplicación.

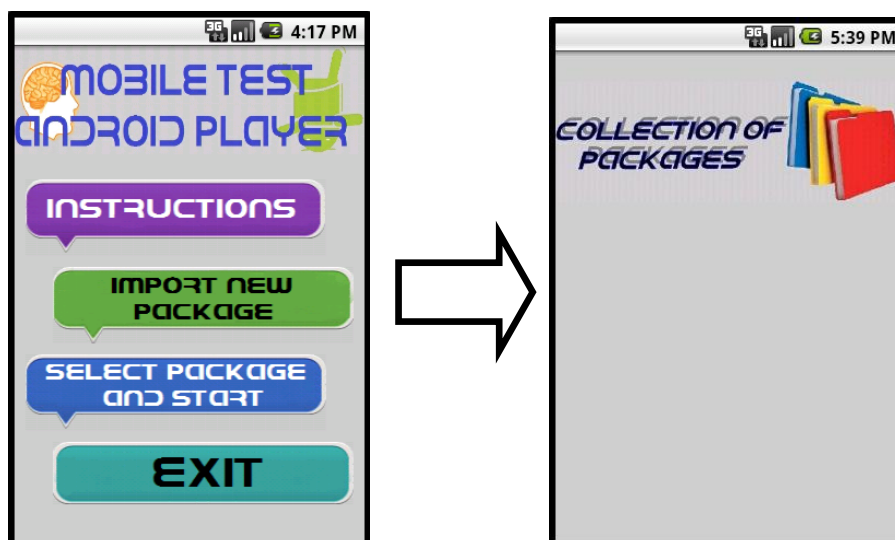


Figura 38. Pantalla Listado de Materiales Educativos

Esta interfaz muestra un listado con el nombre de cada paquete que hay almacenado en la carpeta de la aplicación dentro de la tarjeta de memoria del dispositivo. Inicialmente estará vacía puesto que es la primera vez que se accede y nunca se descargaron archivos. Este paso si bien es recomendable, no es imprescindible, puesto que el usuario conectando el dispositivo móvil a un ordenador puede crear la carpeta de destino con el nombre de la aplicación.

De vuelta al menú principal, desde el mismo, accederemos a la descarga de material educativo pulsando “import new package”. Este nos muestra una interfaz básica con un cuadro de texto donde introducir una dirección web de descarga válida y un botón para aceptar dicha dirección. Para que el usuario sepa en todo momento qué está pasando, la aplicación muestra una vez que pulsa un botón “accept” un pop-up indicando la descarga y una vez finalizada, otro pop-up indicando su descompresión.

Al finalizar dichas operaciones el paquete queda guardado y descomprimido en la carpeta del programa. Así mismo devuelve al usuario al menú principal.

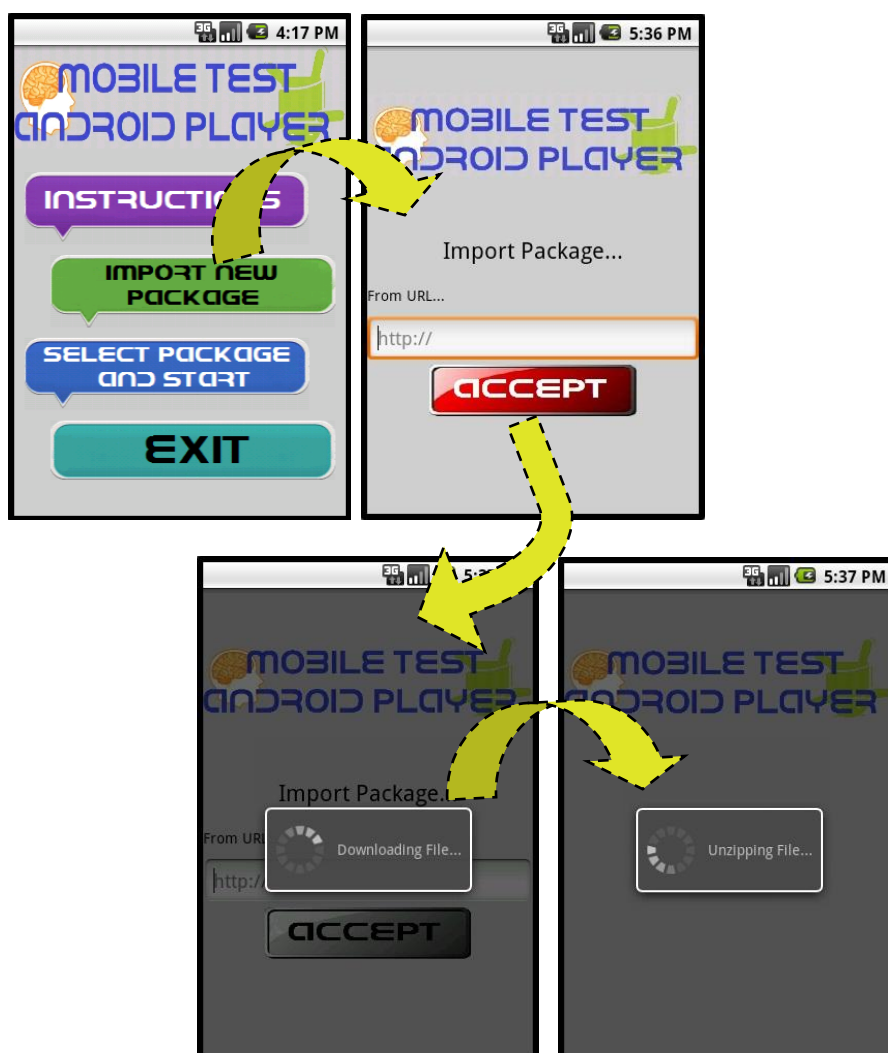


Figura 39. Pantallas Descarga Material Educativo

Si hubiera algún problema con la dirección web (dirección errónea, error de escritura...) la aplicación avisará al usuario.

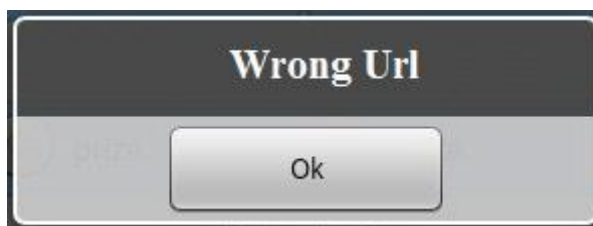


Figura 40. Pantalla de Error

Antes de empezar a realizar los ejercicios, es hora de mirar las instrucciones de la aplicación. Para ello, el usuario pulsará el botón “instructions” del menú principal.

Éstas dan una idea básica del manejo dentro de cada ejercicio para que el usuario no se sienta perdido.

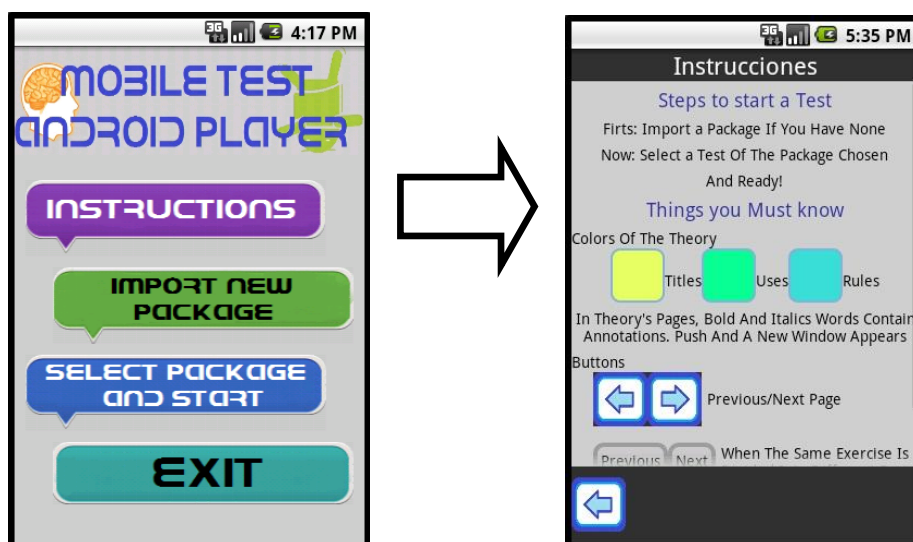


Figura 41. Pantalla Instrucciones

En cualquier momento, el usuario que desee cerrar la aplicación, no tendrá más que pulsar el botón “exit” y ésta terminará su ejecución.

Una vez descargado al menos un paquete con material educativo, el usuario pulsará el botón “select package and star” del menú principal para acceder, primeramente, al listado de paquetes descargados. Elegirá uno de los paquetes pulsando sobre el mismo y accederá a otro listado con los test contenidos en dicho material educativo. Seleccionando uno de ellos comenzará su ejecución.

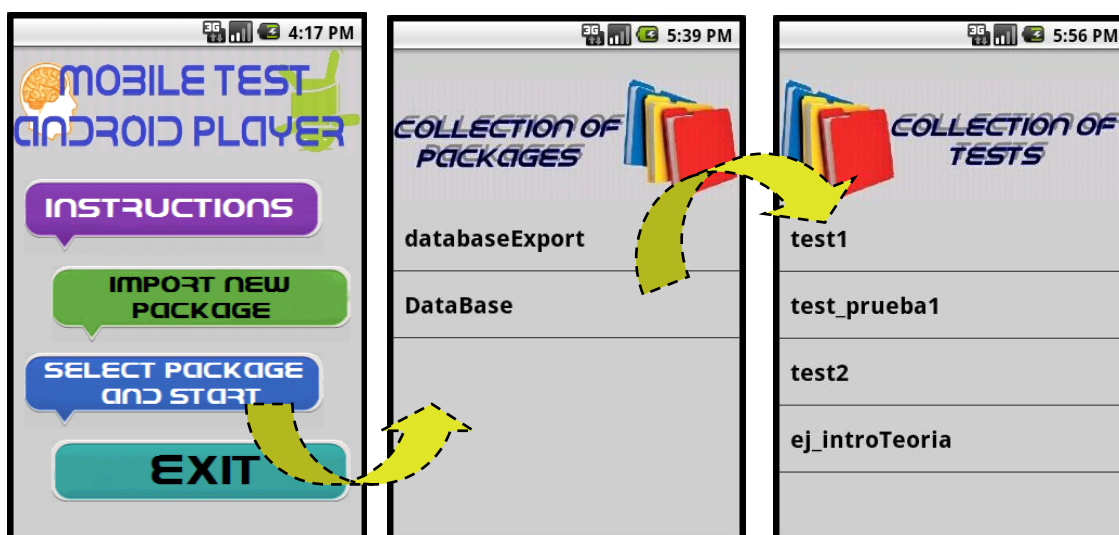


Figura 42. Pantalla Seleccionar Material y Empezar

Si el usuario lo desea, cuando se encuentre en la lista de materiales educativos descargados (“collection of packages”), realizando una pulsación larga sobre cualquiera de los materiales existentes la aplicación le dará la posibilidad de borrar dicha colección.

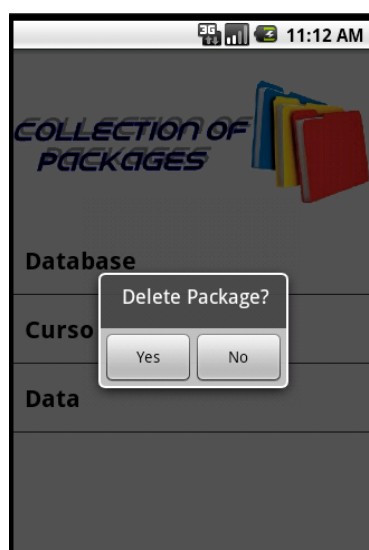


Figura 43. Pantalla Borrar Material

Dentro de cada test podemos tener un número ilimitado de ejercicios, siendo estos de 9 tipos diferentes:

- Introduction
- Theory
- Fill the Gap
- Word Order
- Audio
- Audio Choice
- Drag And Drop
- Multiple Choice
- Multiple Fill the Gap

Antes de adentrarnos dentro de cada tipo de ejercicio, destacamos la barra inferior de la aplicación cuya funcionalidad es común para todos los tipos.

Los botones blancos y azules situados a ambos extremos de la pantalla, son utilizados para movernos de un ejercicio a otro dentro de un mismo test. Si el usuario se encuentra en el primer ejercicio del test, el botón para movernos hacia la izquierda estará deshabilitado. Podrá moverse de ejercicio en ejercicio sin necesidad de responderlo.

En la cara interna de dichos botones, podemos ver otros dos, en este caso de color gris, utilizados para movernos dentro del ejercicio en el caso que este tenga más de una pregunta. Si el ejercicio solo consta de una pregunta, dichos botones estarán ocultos para el usuario.

Por último, el botón central es utilizado para corregir el ejercicio, así el usuario puede ver si la respuesta marcada es correcta o errónea.

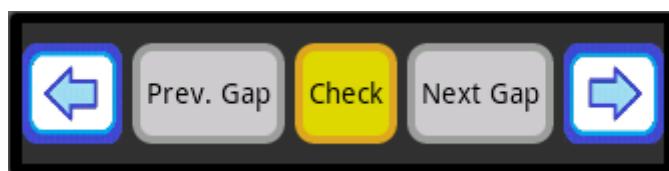


Figura 44. Menú Inferior de Navegación

Si al pulsar el botón “check” no hemos seleccionado ninguna respuesta, se mostrará un mensaje de error indicando el error.

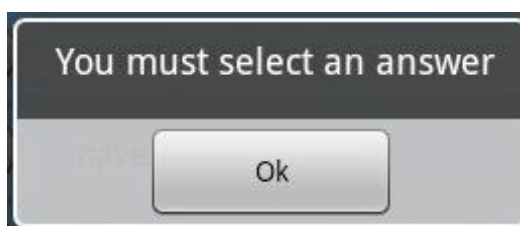


Figura 45. Mensaje de Error Sin Selección

Los ejercicios del tipo “introduction” se caracterizan por tener únicamente texto introductorio de carácter general.

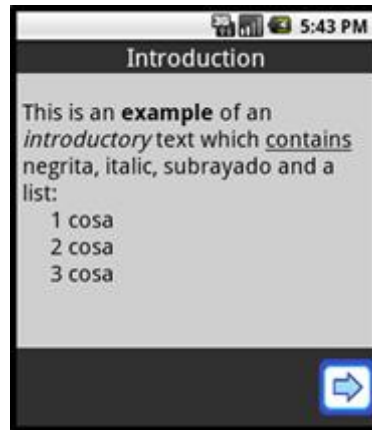


Figura 46. Pantalla Introduction

Los ejercicios del tipo “theory” muestran teoría válida para realizar los ejercicios de los que consta el test. Este tipo de ejercicios tiene 4 tipos de contenido: títulos, usos, imágenes y reglas.



Figura 47. Pantallas Theory

Tanto el contenido de las reglas como el de los usos, puede expandirse para mostrar todo su texto. Pulsando el “+” se expande el contenido, y pulsando “-” se contrae.

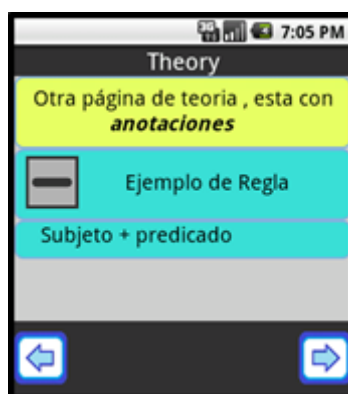


Figura 48. Pantalla Theory Expandir Contenido

Además del contenido comentado anteriormente, también puede mostrar anotaciones. En la interfaz se muestra como texto en cursiva y negrita. El usuario pulsará el texto y podrán ocurrir dos eventos diferentes:

- Si la anotación está contenida en un título, se pulsará dicha anotación y se mostrará el contenido inmediatamente.

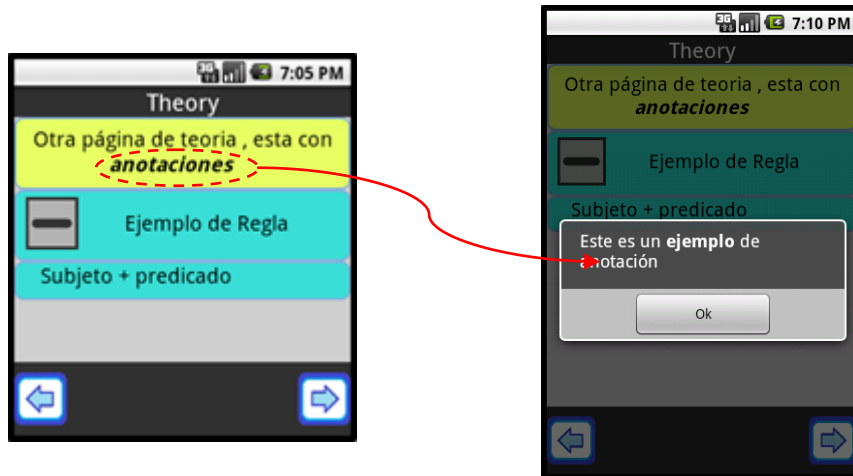


Figura 49. Pantalla Theory Anotaciones

- Si la anotación está contenida en una regla o en un uso, pulsando sobre este, aparece un pop-up indicando que queremos hacer. Si queremos ver la anotación pulsamos “yes”, se mostrará, y al cerrarla veremos el contenido expandido. Si por el contrario no queremos ver la anotación, pasa directamente a expandir el contenido.

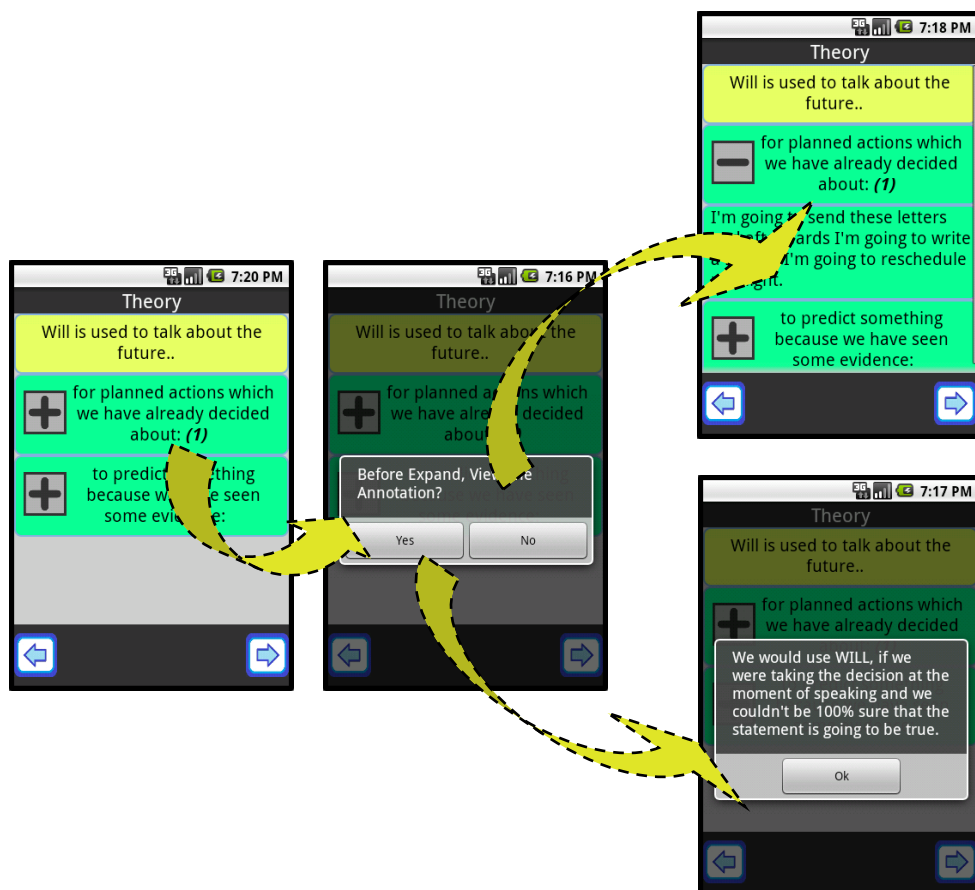


Figura 50. Pantalla Theory Anotaciones + Expandir

En los ejercicios del tipo “fill the gap”, el usuario deberá rellenar el hueco de la pregunta con una de las opciones posibles.

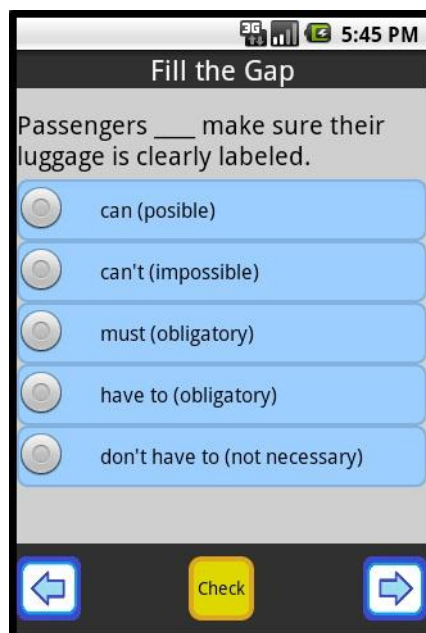


Figura 51. Pantalla Fill The Gap

En los ejercicios del tipo “word order”, el usuario deberá ordenar la frase que muestre el ejercicio introduciendo el texto en el espacio destinado para ello. Deberá respetar el uso de mayúsculas y minúsculas.

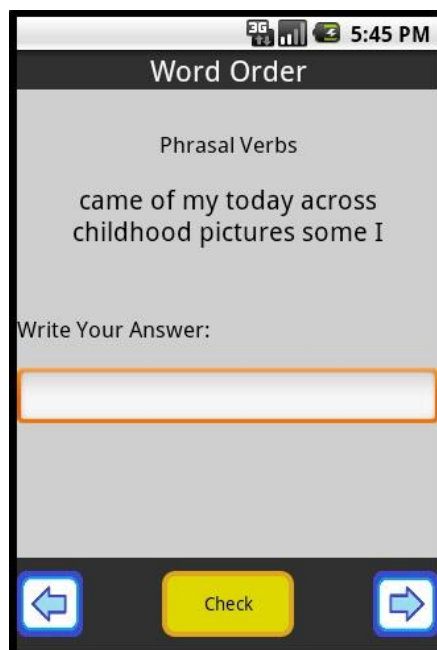


Figura 52. Pantalla Word Order

En los ejercicios “audio” el usuario tendrá disponible un archivo de audio que podrá escuchar cuantas veces quiera. Además se le ofrece un control de volumen para que lo ajuste según sus necesidades.



Figura 53. Pantalla Audio

En los ejercicios “audio choice”, como en los tipo “audio”, se ofrece un archivo de audio y un control de volumen. Además, el usuario deberá responder una serie de preguntas sobre el archivo de audio.



Figura 54. Pantalla Audio Choice

En los ejercicios del tipo “drag and drop”, como un terminal Smartphone es demasiado pequeño para los ejercicios de arrastrar y soltar, se ha optado por que el usuario rellene el hueco como si fuera un ejercicio del tipo “multiple fill the gap”. Con los botones Prev./Next gap podrá moverse por las posibles respuestas de cada uno de los gap de los que conste el ejercicio.

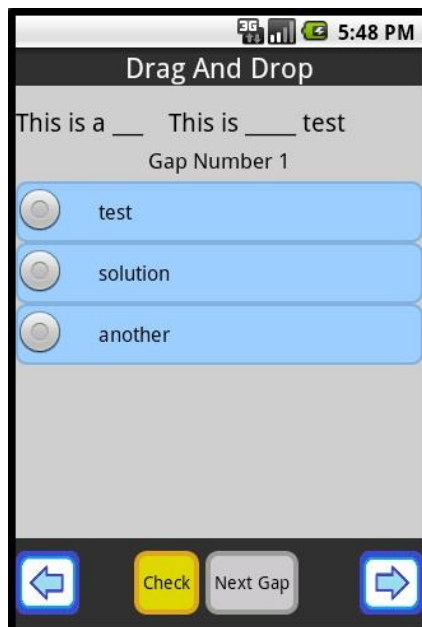


Figura 55. Pantalla Drag And Drop

En el “multiple choice”, el usuario deberá continuar la frase con una de las opciones que se muestran en la interfaz del ejercicio.

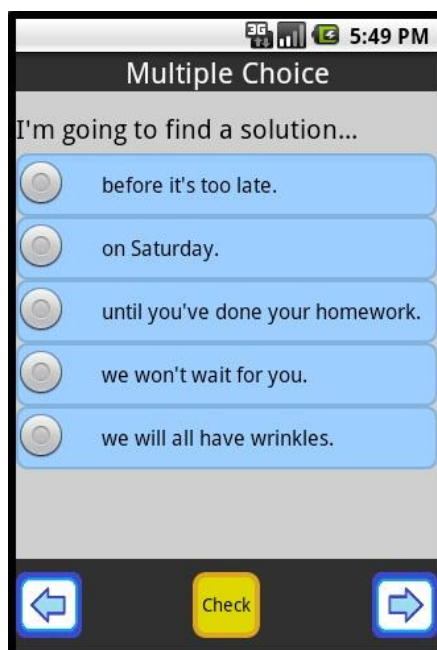


Figura 56. Pantalla Multiple Choice

En los ejercicios del tipo “multiple fill the gap”, la dinámica es la misma que para “drag and drop” o para “fill the gap”, rellenar el/los huecos con las respuestas correctas.

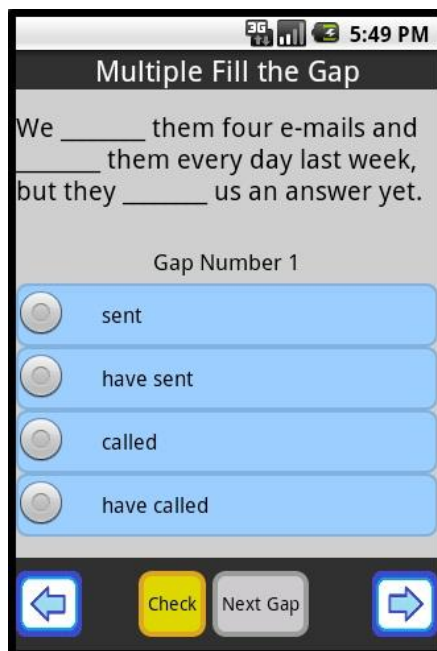


Figura 57. Pantalla Multiple Fill The Gap

Una vez que se llega al último ejercicio del test, la siguiente pantalla que se muestra al usuario tiene como funcionalidad principal, mostrar los resultados y las soluciones. Además podremos reiniciar el test para comenzar de nuevo, volver al menú principal para hacer un nuevo test, o volver atrás en el test por si no hemos contestado todos los ejercicios.



Figura 58. Pantalla Menú Score, Solution And Reset

Si el usuario desea ver los resultados del test realizado, es decir, cuantas preguntas acertó y cuantas ha fallado, deberá pulsar el botón “show score”. Los ejercicios no contestados aparecerán como 0/0. Se ofrece al usuario además, poder volver al menú principal pulsando el botón “home” o retroceder a la pantalla anterior.

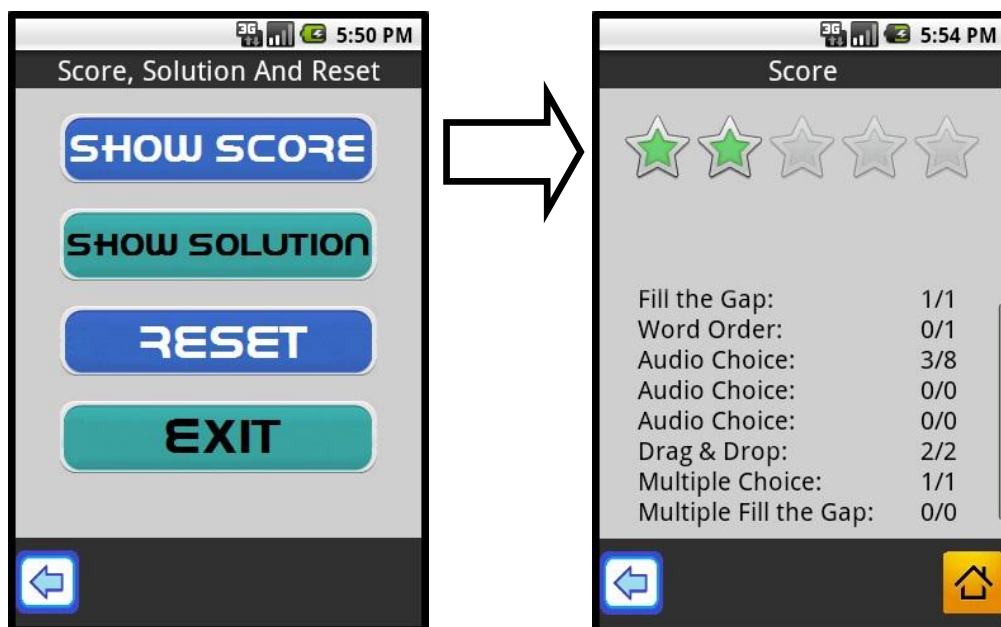


Figura 59. Pantalla Score

En el caso que el usuario desee ver las soluciones, deberá pulsar el botón “show solution”. Destacar que sólo podrá ver las soluciones de los ejercicios contestados. Se ofrece al usuario además, poder volver al menú principal pulsando el botón “home” o retroceder a la pantalla anterior.

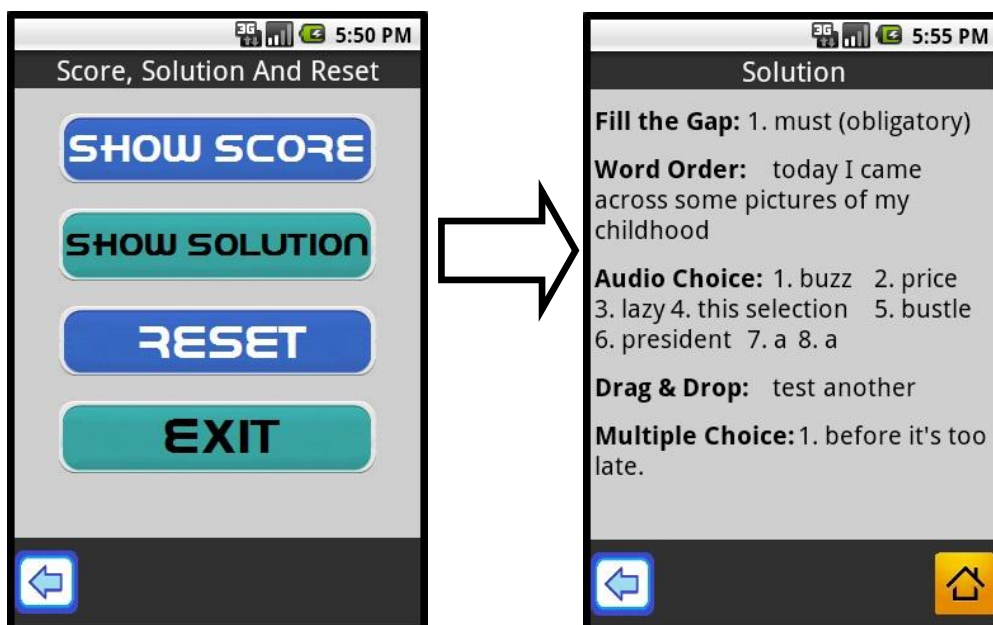


Figura 60. Pantalla Solution

Si pulsamos el botón “reset” el test volverá a comenzar, borrando todos los resultados anteriores.

Al pulsar el botón “exit”, el usuario vuelve al menú principal de la aplicación.

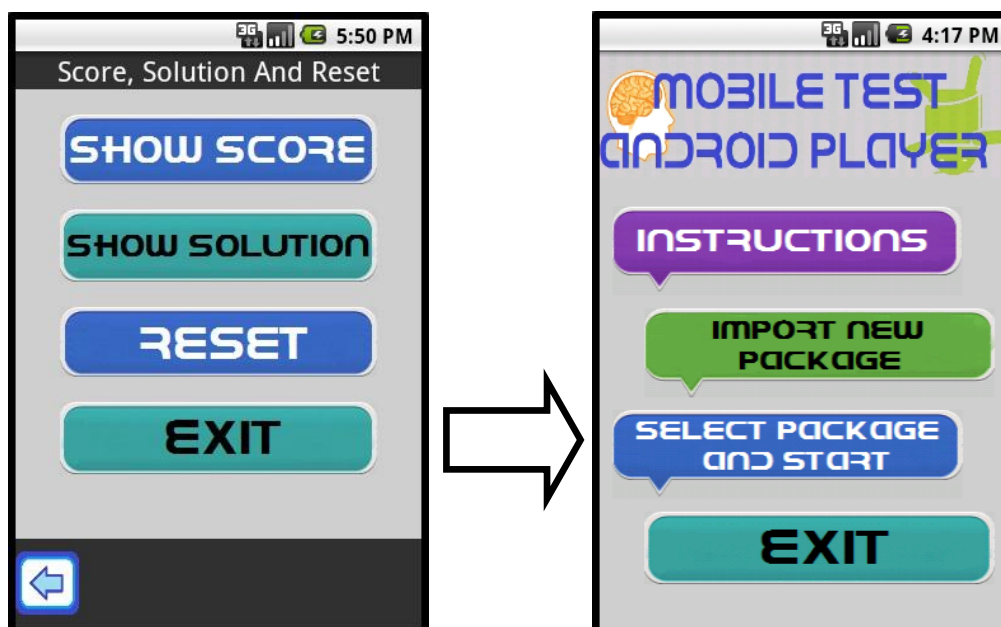


Figura 61. Pantalla Principal

7.2 Manual de Instalación

El objetivo del presente apartado es mostrar las diferentes interfaces que intervienen a la hora de instalar la aplicación en el terminal móvil desde el archivo con extensión apk que lo contiene. De esta forma, obtendremos un pequeño manual de instalación.

Instalar programas a través de sus archivos .APK es útil porque, además de que no existe la necesidad de buscarlo en el Market de Android, se pueden descargar en un ordenador y enviarlo al teléfono a través de una conexión USB, sin la necesidad de usar una conexión a Internet con el coste que supone.

Antes de explicar el proceso de instalación, destacar que es necesario que el dispositivo móvil tenga instalado un administrador de archivos, para poder navegar por el contenido de la tarjeta de memoria. El administrador de archivos usado en la siguiente explicación es ASTRO Administrador de archivos, ubicado en el market de Android y de coste gratuito.

El primer paso es descargar el archivo. APK en el equipo. Conecte el dispositivo móvil a la máquina y transfiera el archivo ya sea en la memoria de este, o en la tarjeta de memoria.

Desconecte el cable USB y abra el administrador de archivos del teléfono.

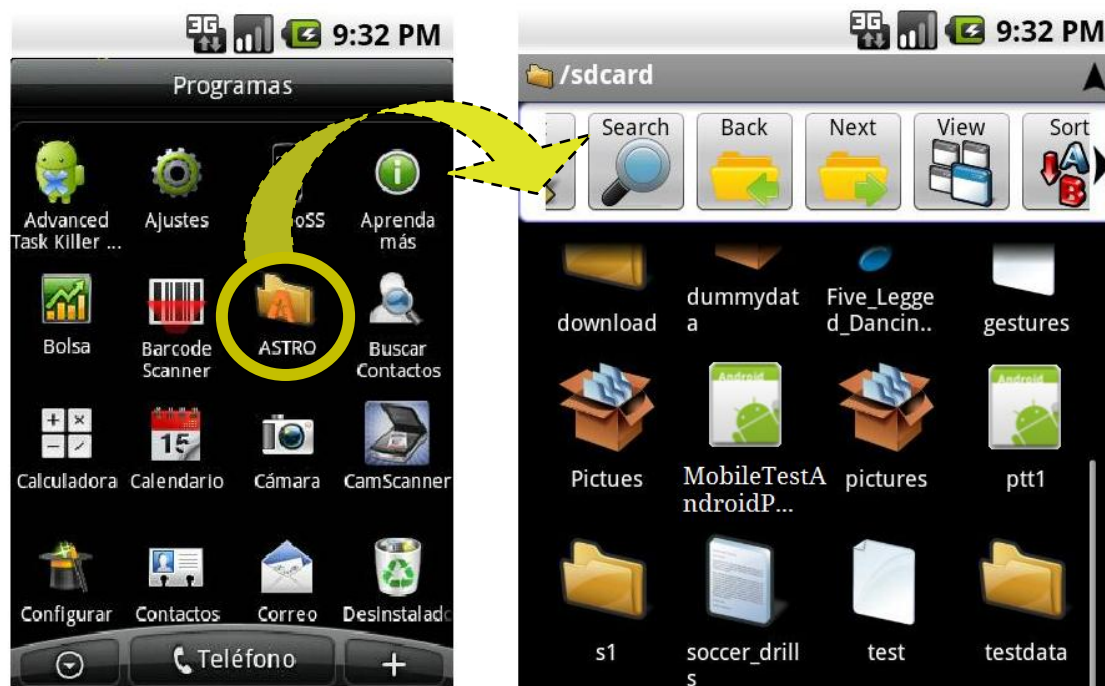


Figura 62. Pantalla Administrador de Archivos

Busque el archivo que acaba de descargarse y pulse sobre él. Seleccione “Abrir el Gestor de Aplicaciones” u “Open App Manager” y a continuación pulse en “Instalar” o “Install”

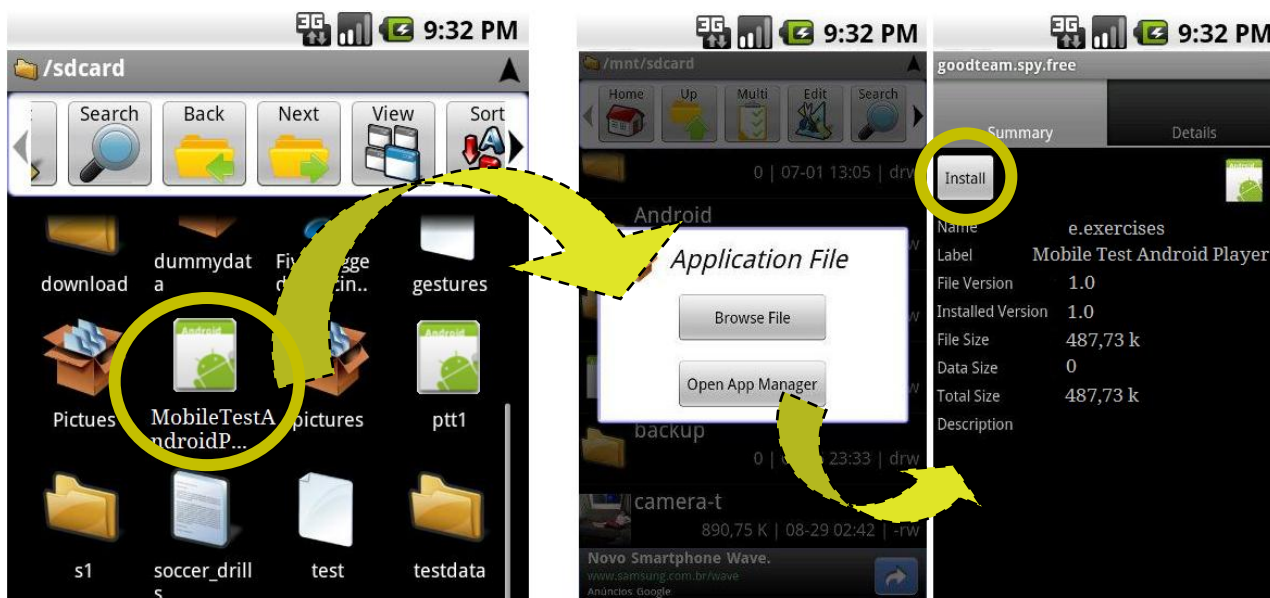


Figura 63. Pantalla Archivo MobileTestAndroidPlayer.apk

La aplicación quedará instalada y lista para usarse.



Figura 64. Pantalla Menú teléfonos Android

Capítulo 8

CONCLUSIONES Y TRABAJOS FUTUROS

8.1 Conclusiones

Uno de los principales motivos por el que decidí realizar este proyecto, era la posibilidad de trabajar con una de las principales plataformas móviles, Android. Me motivó poder aprender sobre un tema que está a la orden del día y en continuo cambio, usado por millones de personas en todo el mundo y que se ha hecho indispensable.

Además de todo lo citado, tenía la motivación extra de no haber estudiado nada acerca sobre el tema, ver hasta dónde puedo llegar, ampliar conocimientos... Ha sido un trabajo duro, pero muy gratificante. El hecho de ver que en lo que llevas trabajando meses, por fin funciona, es una gran sensación.

Por lo tanto, el primer punto positivo de este proyecto es haberlo conseguido terminar con todas las funcionalidades acordadas y en correcto funcionamiento. Poder aportar algo nuevo y útil a la comunidad universitaria, ver que tu esfuerzo sirve para que otros alumnos aprendan de forma entretenida con tu herramienta.

Otro punto importante, más para mi persona, es el poder haberme adentrado en el mundo de las plataforma móviles, a día de hoy y gracias al proyecto con Android, dando pío al día de mañana a conocer alguna otra. Esto me permitirá tener una ventaja competitiva en el mundo laboral.

Con respecto a la plataforma Android, el hecho de ser código abierto facilita mucho la labor del desarrollador. Pero sobre todo, lo que más me atrae es la libertad que tiene. Es decir, puedes hacer una aplicación capaz de usar cada componente del Smartphone, cada funcionalidad, puedes mejorar el software, cambiar widgets, launcher... y además si eres propietario de un terminal con Android le puedes instalar lo que te venga en gana. Estas características, desde mi punto de vista, dan una ventaja competitiva a la plataforma.

Siguiendo el hilo, no podría haber realizado la aplicación sin Java, lenguaje que he tenido que aprender puesto que en la carrera no se vio nada. Lejos de ser un lastre el no conocer el lenguaje, quizás ha sido una ventaja, puesto que me ha permitido ir adquiriendo conocimientos a la par que aprendía a usar el entorno de desarrollo de Android. Poco a poco, enlazando ideas, consultando manuales, foros... iban surgiendo funcionalidades. Otra apartado más en los lenguajes de programación de mi curriculum.

Llegados a este punto, valorar el resultado final del proyecto MobileTestAndroidPlayer. El hecho de tener claro desde un principio qué es lo que debe hacer la aplicación facilitó mucho el trabajo, seguir unas pautas de estilo y contenido ahorró mucho tiempo en el desarrollo. El objetivo propuesto de mostrar toda la funcionalidad acordada se ha cumplido, incluso incluyendo algo más para darle un toque más distintivo a la aplicación como puede ser el control de volumen.

En cuanto a la interfaz, el objetivo era crear un diseño que fuera capaz de atraer al público universitario, algo que no fuera monótono. Creo que, aunque siempre puede hacerse mejor, la interfaz principal del programa es atractiva a la vista, y los ejercicios

tienen todos colores llamativos para atraer la atención. Por lo tanto, ese objetivo queda cubierto.

En conclusión, estoy muy satisfecho por haber cumplido todos los objetivos marcados, el trabajo de tantos meses ha dado su fruto. Además, he cumplido mi objetivo personal. Aprender y desenvolverme con soltura en un corto periodo de tiempo tanto con el lenguaje Java como en la plataforma de desarrollo de Android Eclipse, lo que me abre una puerta al mundo laboral que en estos momentos es de agradecer.

8.2 Trabajos Futuros

Como se ha mencionado anteriormente, nada es perfecto. Todo puede mejorarse. Para ello, se pueden incluir nuevas funcionalidades o modificaciones de las ya existentes.

Así pues podemos enumerar algunos de los trabajos futuros que se pueden contemplar:

- **Guardar Histórico de Puntuaciones:** Actualmente la aplicación guarda el resultado del test realizado hasta que se inicia uno nuevo, momento en el que borra todo para recoger los resultados del siguiente. Lo que se propone es crear una herramienta que guarde el nombre del test y las calificaciones obtenidas.
- **Ampliar Para Otros Idiomas:** Aunque el diseño de la aplicación se ha realizado de forma genérica para que funcione con materiales educativos cuyo contenido esté en cualquier idioma, los botones, mensajes, texto propio de la aplicación... todo está en inglés. Se propone adaptar esa pequeña porción de código para que dependiendo del idioma, cambie también el idioma del texto de botones, mensajes...
- **Estudio de la Apariencia:** La apariencia de un programa puede determinar la elección del usuario acerca de su uso. Por lo tanto se propone hacer un estudio sobre los gustos y necesidades de los usuarios potenciales (pudiendo mostrarles un prototipo, etc.) para dotar a la aplicación de la apariencia deseada para incrementar su uso.

Capítulo 9

GESTIÓN DEL PROYECTO

En el presente capítulo se desarrolla la planificación seguida para la elaboración del proyecto, apoyada en un diagrama Gantt, y se detallan los costos asociados al desarrollo de la aplicación mediante la elaboración de un presupuesto.

9.1 Planificación

A continuación se detalla la planificación inicial, acordada en la primera reunión, para la elaboración del proyecto.

En primer lugar se muestra la tabla con las fechas y duración de cada tarea, todas ellas estimadas.

Nombre	Fecha de inicio	Fecha de fin	Duración
Estudio Inicial	1/09/11	15/09/11	10
Prototipo	15/09/11	22/09/11	5
[-] Análisis	22/09/11	30/09/11	6
Casos de Uso	22/09/11	27/09/11	3
Requisitos	27/09/11	30/09/11	3
[-] Diseño	30/09/11	8/10/11	6
Diseño General	30/09/11	5/10/11	3
Diagramas de Secuencia	5/10/11	8/10/11	3
[-] Implementación	10/10/11	31/12/11	60
Lectura XML	10/10/11	29/10/11	15
Descarga Zip	31/10/11	12/11/11	10
Listado Carpeta	14/11/11	26/11/11	10
Funcionalidades Ejercicios	28/11/11	24/12/11	20
Interfaz	26/12/11	31/12/11	5
[-] Pruebas	31/10/11	24/12/11	40
Prueba Lectura XML	31/10/11	2/11/11	2
Pureba Descarga Zip	14/11/11	16/11/11	2
Prueba Listar Contenido	28/11/11	30/11/11	2
Prueba Funcionalidades	30/11/11	24/12/11	18
Documentación	7/10/11	31/01/12	82

Figura 65. Estimación inicial: Fechas y Duración

Las tareas más complejas se han dividido en tareas más simples.

En la siguiente figura podemos ver el diagrama de Gantt con todas y cada una de las fases citadas anteriormente. Se puede observar que las tareas de Implementación, Pruebas y Documentación se solapan en el tiempo puesto que pueden realizarse al mismo tiempo.

Por último, existen dependencias entre todas y cada una de las tareas (exceptuando la Documentación) puesto que no se puede comenzar su realización sin haber finalizado la tarea anterior.

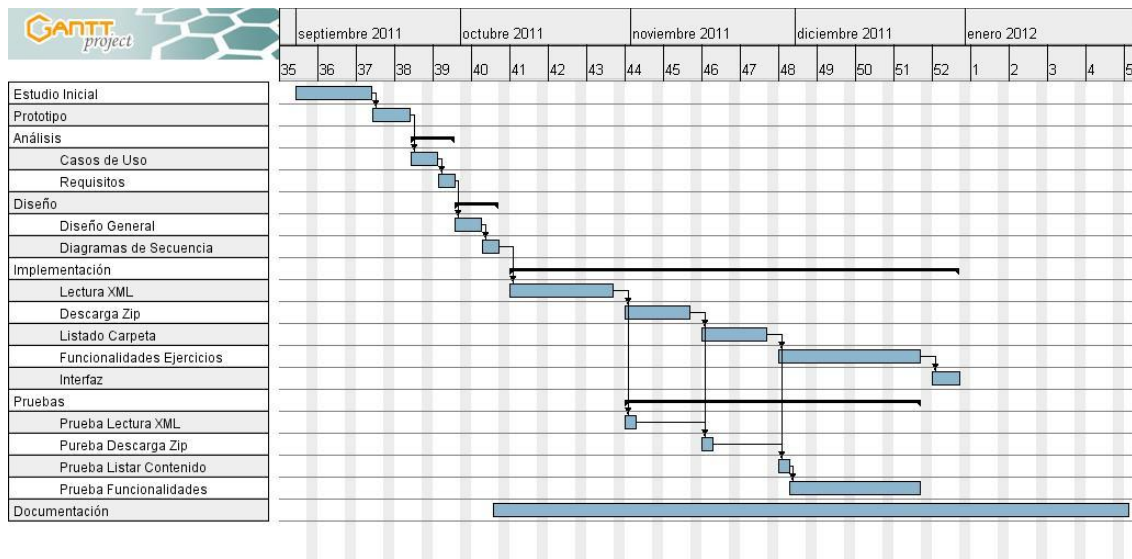


Figura 66. Gráfica Gantt Estimación inicial

Las tareas Estudio Inicial y Prototipo corresponden a una labor de investigación y estudio sobre la plataforma Eclipse de Android y el lenguaje de programación Java, ambos desconocidos hasta la fecha de inicio del proyecto.

Por lo tanto, la duración total estimada para la elaboración del proyecto es de **108 días**, con una dedicación de **4 horas al día** aproximadamente.

Una vez llegados a la finalización del proyecto, se realiza otra estimación esta vez con el tiempo real invertido.

Nombre	Fecha de inicio	Fecha de fin	Duración
Estudio Inicial	13/09/11	24/09/11	9
Prototipo	26/09/11	7/10/11	9
[-] Análisis	7/10/11	14/10/11	5
Casos de Uso	7/10/11	8/10/11	1
Requisitos	10/10/11	14/10/11	4
[-] Diseño	14/10/11	21/10/11	5
Diseño General	14/10/11	15/10/11	1
Diagramas de Secuencia	17/10/11	21/10/11	4
[-] Implementación	21/10/11	22/12/11	44
Lectura XML	21/10/11	5/11/11	11
Descarga Zip	7/11/11	12/11/11	5
Listado Carpeta	14/11/11	19/11/11	5
Funcionalidades Ejercicios	21/11/11	17/12/11	20
Interfaz	19/12/11	22/12/11	3
[-] Pruebas	7/11/11	17/12/11	30
Prueba Lectura XML	7/11/11	9/11/11	2
Pureba Descarga Zip	14/11/11	16/11/11	2
Prueba Listar Contenido	21/11/11	23/11/11	2
Prueba Funcionalidades	23/11/11	17/12/11	18
Documentación	7/10/11	22/02/12	98

Figura 67. Estimación real: Fechas y Duración

Podemos observar tanto en la tabla de fechas como en el diagrama de Gantt, que se retrasa la fecha de inicio del proyecto. Esto se debe a la realización de un examen. Además, se acorta el plazo de Implementación en 16 días puesto que se consiguen realizar los objetivos en un menor plazo de tiempo y se alarga la fase de Documentación debido a un problema de redacción en los primeros temas que implicó rehacerlos de nuevo.

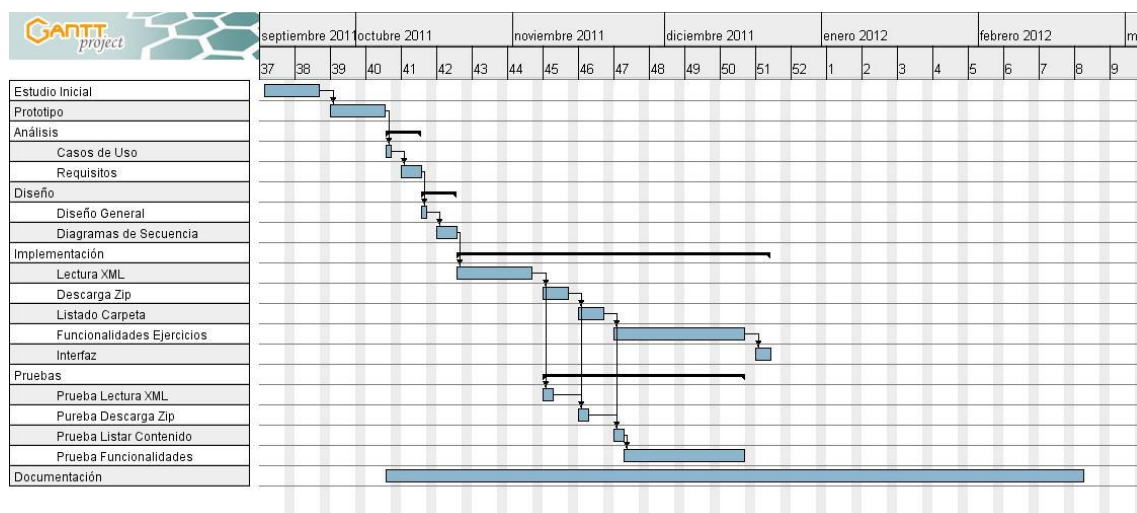


Figura 68. Gráfica Gantt Estimación real

Por lo tanto, la duración total real para la elaboración del proyecto es de **116 días**, con una dedicación de **4 horas al día** aproximadamente. Se han necesitado solamente **8 días** más de trabajo para cumplir con los objetivos. Sin embargo, se ha sufrido un gran retraso en la finalización del mismo puesto que se estimo finalizarlo el **31 de enero de 2012** y la fecha real ha sido el **22 de febrero de 2012**.

9.2 Presupuesto

Establecidas la planificación y duración del proyecto, en el apartado actual se detallan todos los gastos relacionados para la realización del proyecto. Estos serán gastos de personal, equipo y otros costes, siguiendo como guía la plantilla que proporciona la universidad para ello [34].

Conociendo la fórmula

$$\text{Coste} = ((\text{duración días} * \text{horas diarias}) / \text{dedicación hombre mes}) * \text{coste hombre mes}$$

Se calcula los gastos de personal, teniendo en cuenta que:

- Duración días = 116
- Horas diarias = 4
- Dedicación hombre mes = 131,25
- Coste hombre mes = 2694,39

Dando como resultado unos gastos de personal de **NUEVE MIL QUINIENTOS VEINTICINCO EUROS CON TREINTA Y UN CÉNTIMOS DE EUROS**.

Para la elaboración del proyecto ha sido necesario adquirir:

- Netbook Hp Mini 110-3120ss a un coste de 299 euros.
- Smartphone Orange Boston con un coste de 120 euros.

Siguiendo la plantilla, se calcula la amortización para los equipos.

Descripción	Coste (Euro)	% Uso dedicado proyecto	Dedicación (meses)	Periodo de depreciación	Coste imputable ^{d)}
Netbook Hp	299,00	100	5,8	60	28,90
Orange Boston	120,00	100	5,8	60	11,60
Total					40,50

Tabla 83. Amortización de equipos

^{d)} Fórmula de cálculo de la Amortización:

$$\frac{A}{B} \times C \times D$$

A = nº de meses desde la fecha de facturación en que el equipo es utilizado

B = periodo de depreciación (60 meses)

C = coste del equipo (sin IVA)

D = % del uso que se dedica al proyecto (habitualmente 100%)

Para finalizar con el presupuesto, se detalla a continuación el resumen de costes:

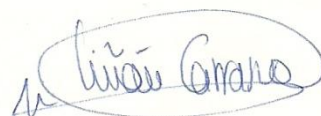
Presupuesto Costes Totales	Presupuesto Costes Totales
Personal	9.525,31
Amortización	40,50
Costes Indirectos (20%)	1.913,16
Total Sin IVA	11478,97
Total Con IVA (18%)	13545,19

Tabla 84. Resumen Costes

El presupuesto total del proyecto ascienda a la cuantía de **TRECE MIL QUINIENTOS CUARENTA Y CINCO EUROS CON DIECINUEVE CÉNTIMOS DE EURO.**

Leganés, a de de 2012,

El ingeniero proyectista



Fdo. Alfonso Liñán Carrasco

Anexo A

MOBILE TEST CREATOR

Mobile Test Creator se trata de una herramienta de creación de páginas (ejercicios) y tests para su uso en la herramienta móvil que se desarrolla en el presente documento, **MobileTestAndroidPlayer**.

Esta herramienta será usada por el cuerpo docente, quienes crearán los materiales educativos con el contenido que deseen para posteriormente exportar dichos materiales en formato ZIP a la herramienta móvil.

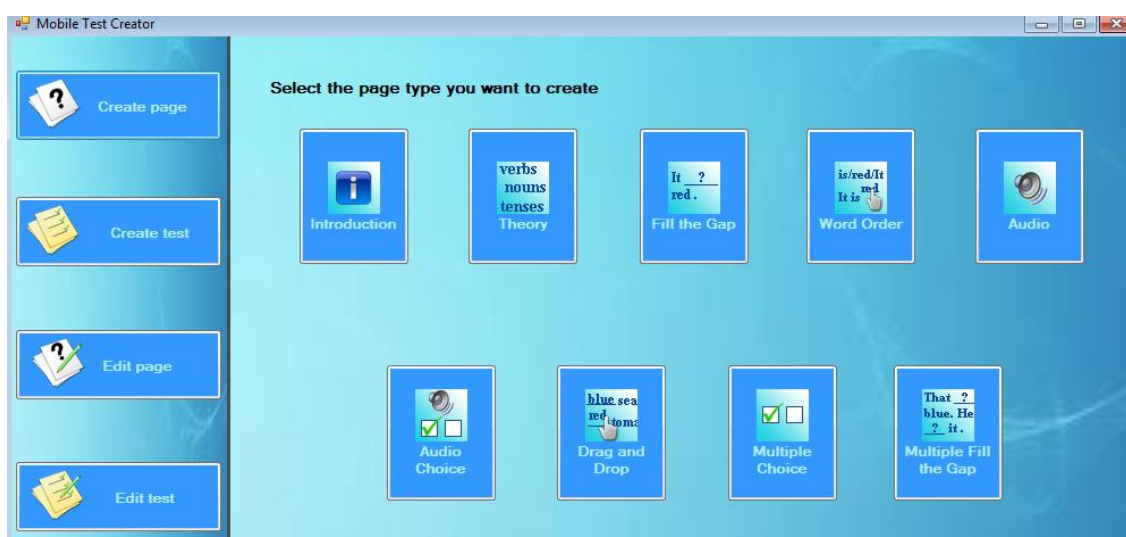


Figura 69. Pantalla Creación de Páginas de MobileTestCreator

Primero, los educadores crearan tantas páginas (ejercicios) como deseen y de los tipos que gusten (siendo estos los mismos que reconocerá nuestra aplicación). Aquí deberán introducir, en el caso que sea necesario, los archivos de audio e imágenes que la aplicación tendrá que reproducir.

A continuación, creará el o los tests que crean convenientes. Para ello introducirán un nombre para denominar al test y posteriormente añadirán la/s página/s de las que constará dicho test.

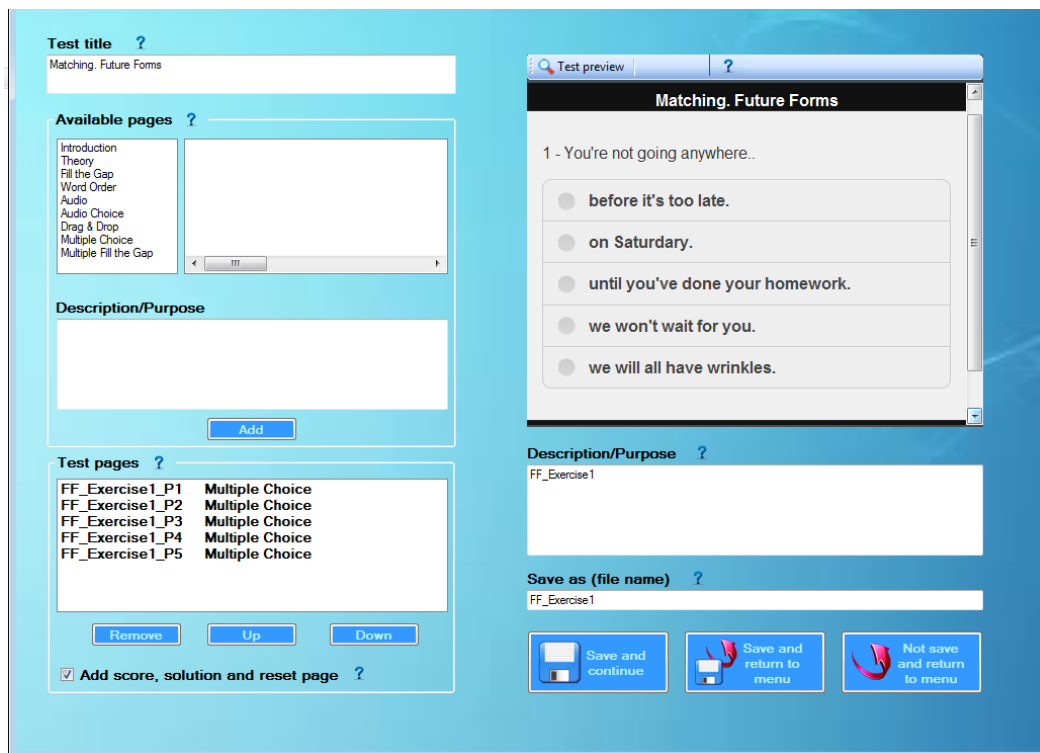


Figura 70. Pantalla Creación de Test de MobileTestCreator

Por último, el educador exportará el contenido para ser usado por nuestra aplicación. Se generará un archivo Zip con todas las páginas y tests de la aplicación así como los recursos necesarios para su funcionamiento.

GLOSARIO

TÉRMINO	DESCRIPCIÓN
Sd Card	Secure Digital Card
UML	Unified Modeling Language
SDK	Software Development Kit
WAP	Wireless Application Protocol
PDA	Personal Digital Assistant
SO	Sistema Operativo
OS	Operating System
API	Application Programming Interface
ROM	Read-Only Memory
APP	Application
USB	Universal Serial Bus
IEEE	Institute of Electrical and Electronics Engineers
URL	Uniform Resource Locator
APK	Application Package

Tabla 85. Glosario de Términos

REFERENCIAS

- [1] Mendoza, Jose A. “e-Learning, el futuro de la educación a distancia” [en línea] 2003. <http://www.informaticamilenium.com.mx/paginas/mn/articulo78.htm> (7/12/2011)
- [2] Cabero, Julio. “Bases pedagógicas del e-learning” [en línea] Revista de Universidad y Sociedad del Conocimiento Vol. 3 - N.º 1 / Abril de 2006 <http://www.uoc.edu/rusc/3/1/dt/esp/cabero.pdf> (3/12/2011)
- [3] Rubio Gómez, Celia. González moreno, Vanessa. Selas Felguera, Mª José. “M-LEARNING” [en línea] s.f. <http://tice.wikispaces.com/m-learning> (7/12/2011)
- [4] ISEA S.Coop. “MOBILE LEARNING, Análisis prospectivo de las potencialidades asociadas al Mobile Learning” [en línea] 2009. http://www.iseamcc.net/eISEA/Vigilancia_tecnologica/informe_4.pdf (5/12/2011)
- [5] Equihua, Saul. “M-learning: Algunas consideraciones” [en línea] 2011. <http://www.biblogtecarios.es/saulequihua/m-learning-algunas-consideraciones> (7/12/2011)
- [6] neospacian. “7 Pasos Para El Desarrollo De Un Programa” [en línea] 2011. <http://www.buenastareas.com/ensayos/7-Pasos-Para-El-Desarrollo-De/2740647.html> (7/12/2011)
- [7] Arturo Baz Alonso, Irene Ferreira Artime, María Álvarez Rodríguez, Rosana García Baniello “Dispositivos Móviles” [en línea] s.f. <http://156.35.151.9/~smi/5tm/09trabajos-sistemas/1/Memoria.pdf> (9/12/2011)

- [8] Comisión del Mercado de las Telecomunicaciones. “NOTA MENSUAL Mayo-2011” [en línea] 2011. http://www.cmt.es/es/publicaciones/anexos/110705_NM_mayo2011.pdf (9/12/2011)
- [9] Joss Gillet. “Wireless Intelligence: Global mobile connections to surpass 6 billion by year-end” [en línea] 2011. <https://www.wirelessintelligence.com/analysis/pdf/2011-09-08-global-mobile-connections-to-surpass-6-billion-by-year-end.pdf> (9/12/2011)
- [10] Jaime Aranaz Tudela. “DESARROLLO DE APLICACIONES PARA DISPOSITIVOS MÓVILES SOBRE LA PLATAFORMA ANDROID DE GOOGLE” [en línea] 2009. http://e-archivo.uc3m.es/bitstream/10016/6506/1/PFC_Jaime_Aranaz_Tudela_2010116132629.pdf (9/12/2011)
- [11] Arturo Baz Alonso, Irene Ferreira Artime, María Álvarez Rodríguez, Rosana García Baniello “Dispositivos Móviles” [en línea] s.f. <http://156.35.151.9/~smi/5tm/09trabajos-sistemas/1/Memoria.pdf> (9/12/2011)
- [12] StatCounter GlobalStats. [en línea] <http://gs.statcounter.com/> (14/12/2011)
- [13] Angel Vilchez. “Que es Android: Características y Aplicaciones” [en línea] 2009. <http://www.configurarequipo.com/doc1107.html> (10/12/2011)
- [14] Apple, ios dev center. [en línea] <http://www.apple.com/es/iphone/ios/> (11/12/2011)
- [15] W-Bot. “Symbian OS, Definición e Historia” [en línea] 2010. <http://comunidad.wilkinsonpc.com.co/symbian-os-265/symbian-os-definicion-e-historia-15296.html> (11/12/2011)
- [16] Universidad de Málaga. “SYMBIAN OS C++ ¿QUÉ ES SYMBIAN?” [en línea] s.f. <http://www.lcc.uma.es/~pedro/mobile/Projects/pdfs/Tutorial%20symbian.pdf> (11/12/2011)
- [17] juanjprince. “OS MÓVILES: Android, Symbian, Blackberry y muchos mas...” [en línea] 2010. <http://jjprince.wordpress.com/2010/11/08/os-moviles-android-symbian-blackberry-y-muchos-mas/> (11/12/2011)
- [18] Nelson Aguilar. “Sistema Operativo para teléfonos móviles diseñados por Microsoft” [en línea] 2011. <http://esistemasoperativos.mforos.com/1834161/10207688-telefonos-moviles-de-windows/> (11/12/2011)
- [19] Javier Penalva. “Windows Phone 7” [en línea] 2010. <http://www.xataka.com/moviles/windows-phone-7> (13/12/2011)
- [20] openBoxer. “Lenguaje de programación Java” [en línea] 2010. <http://www.openboxer.260mb.com/java.php> (14/12/2011)

- [21] Bartolomé Sintés Marco. “Qué es el XML” [en línea] 2011. http://www.mclibre.org/consultar/xml/lecciones/xml_quees.html (14/12/2011)
- [22] Francisco Mendoza. “Introducción a XML (Extensible Markup Language)” [en línea] s.f. <http://www.ilustrados.com/tema/3783/Introduccion-Extensible-Markup-Language.html> (14/12/2011)
- [23] Condesa. “Usando layouts basadas en XML con Android” [en línea] 2011. <http://androideity.com/2011/07/11/usando-layouts-basadas-en-xml-con-android/> (14/12/2011)
- [24] Jose Mendiola. “El número de apps gratuitas de Android ya supera al de iPhone” [en línea] 2011. <http://es.engadget.com/tag/Distimo/> (16/12/2011)
- [25] PcActual. “Más de la mitad de las aplicaciones Android son gratuitas” [en línea] 2010. http://www.pcactual.com/2010/07/09/1788/mas_mitad_las_aplicaciones_android_son_gratuitas.html (16/12/2011)
- [26] Christy Pettey, Holly Stevens. “Gartner Says Android to Command Nearly Half of Worldwide Smartphone Operating System Market by Year-End 2012” [en línea] 2011. <http://www.gartner.com/it/page.jsp?id=1622614> (17/12/2011)
- [27] Android Developers “What is Android?” [en línea] 2012 <http://developer.android.com/guide/basics/what-is-android.html> (07/02/2012)
- [28] Alejandro. “Interfaz de usuario en androide” [en línea] 2010 <http://blogdeprogramacion.blogspot.com/2010/10/interfaz-de-usuario-en-androide.html> (07/02/2012)
- [29] sgoliver “Estructura de un proyecto Android” [en línea] 2010 <http://www.sgoliver.net/blog/?p=1278> (07/02/2012)
- [30] Lawprier “DESARROLLAR APLICACIONES PARA DISPOSITIVOS MÓVILES CON SISTEMA OPERATIVO ANDROID. CAPÍTULO 2: TU PRIMERA APLICACIÓN EN JAVA PARA ANDROID” [en línea] 2011 <http://www.paratuandroid.com/Desarrollo/desarrollar-aplicaciones-para-dispositivos-moviles-con-sistema-operativo-android-capitulo-2-tu-primera-aplicacion-en-java-para-android.html> (07/02/2012)
- [31] Julián Urbano, Gonzalo Génova, Juan Lloréns, Maricruz Valiente, Diego Martín, Isidro Hernánz, Eduardo Barra, Mónica Marrero, Jorge Morato, Sonia Sánchez, Vicente Palacios, Manuela Alejandres, Omar Hurtado, José Miguel Fuentes. “Modelado Básico con Casos de Uso” [en línea] s.f. http://ocw.uc3m.es/ingenieria-informatica/disenio-de-software-avanzado/material-de-clase-1/04-Modelado_Basico_con_Casos_de_Uso.pdf/view (2/01/2012)

- [32] Joaquín Gracia. “UML: Casos de Uso. Use case Desarrollo de Software Orientado a Objetos” [en línea] 2003 <http://www.ingenierosoftware.com/analisisydiseno/casosdeuso.php> (2/01/2012)
- [33] James & Suzanne Robertson. “Volere Plantilla de Especificación de Requisitos” [en línea] 2006 http://www.volere.co.uk/pdf%20files/template_es.pdf. (2/01/2012)
- [34] Universidad Carlos III de Madrid “Plantilla presupuesto del proyecto fin de carrera ” [en línea] http://www.uc3m.es/portal/page/portal/administracion_campus_leganes_est_cg/proyecto_fin_carrera